h e g

Haute école de gestion
Genève

Une application de co-investissement utilisant la blockchain Stellar

Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Amir SAIDI

Conseillers au travail de Bachelor :

Jean-Philippe TRABICHET

Athanasios PRIFTIS

Genève, le 20.10.2017

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de Gestion

Hes·so GENÈVE
Haute Ecole Spécialisée
de Suisse occidentale

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre Bachelor of Science HES-SO en Informatique de Gestion.

L'étudiant atteste que son travail a été vérifié par un logiciel de détection de plagiat.

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 20.10.2017

Amir SAIDI

Remerciements

Je souhaite remercier m. Athanasios Priftis qui m'a suivi et conseillé tout le long du projet, il m'a aidé à prendre des décisions et à développer les idées sur le projet. Il m'a également aidé à résoudre certains problèmes que j'ai eus et à y trouver des solutions.

Je remercie également m. Jean-Philippe Trabichet, qui m'a accordé confiance en associant son nom à mon travail de bachelor et qui m'a donné des conseils sur le projet et ma présentation.

Je remercie également la HEG, pour nous mettre à dispositions du matériel et des informations utiles pour effectuer correctement nos projets et recherches.

Résumé

Ce document explique brièvement ce qu'est le DAO, analyse une partie de son fonctionnement et analyse et explique la faille qui a été exploitée. Cela permet de se préparer au développement d'un site internet avec des similitudes, en voyant les risques qu'il y a et comment les éviter.

Stellar et Cowaboo sont également décrit brièvement, car ils sont liés au site internet qui a été développé.

Ensuite, différents scénarios sont énumérés pour permettre de choisir le meilleur scénario à développer en fonction de leur fonctionnement et de leurs avantages et inconvénients.

Puis finalement s'en suit, l'explication de l'application, avec les difficultés rencontrées et les solutions apportées.

Table des matières

1. Introduction

En cours de Wikinomie 646-2 enseigné par m. Athanasios Priftis, les élèves devaient développer un prototype d'application dans le langage voulu en utilisant l'API de Cowaboo. Ce développement devait se faire en groupe, donc l'idée de l'application a été développer avec un Jimmy Paris, un camarade de classe.

M. Priftis trouvant l'idée intéressante, proposa de continuer le développement en tant que travail de Bachelor, en y incluant les investissements directement avec une crypto-monnaie. Car de nos jours, la technologie blockchain et les crypto-monnaie sont des sujets qui émergent de plus en plus et c'est un sujet qui a été extrêmement abordé durant la 3ème année de la HEG en Informatique de Gestion, lors de l'année académique 2016/2017.

THE DAO faisant à peu près la même chose, il fallait comprendre ce que c'est THE DAO, comment il fonctionne et surtout, analyser sa faille qui à provoquer un gros changement pour la blockchain Ethereum.

Plusieurs scénarios ont été réfléchis et sont énuméré, pour pouvoir les comparer, voir les avantages et inconvénients de chacun et pouvoir ensuite choisir le plus intéressant pour le projet.

Le but de cette première partie est de bien définir le cadre du projet et ainsi ne pas se lancer dans le développement tête dans le clavier. Cela permet d'anticiper les futurs problèmes, organiser le développement et donc éviter un maximum d'imprévu qui pourrait entraîner l'obsolescence de certaines fonctionnalités déjà développées.

Lors de la 2ème partie, avant de se lancer dans le développement, une planification a été mise en place, en listant toutes les fonctionnalités qui allait être développées, avec leur degré de priorité et leur difficulté.

Une fois la planification faite, les fonctionnalités ont été développées dans l'ordre de leur priorité, distribuées dans 4 périodes de développement, pour permettre une meilleure visualisation de ce qui a été fait et de ce qui reste à faire.

Durant tout le travail de Bachelor, il y a eu plusieurs réunions de suivi, qui ont permis d'analyser les problèmes, les opportunités et ainsi mieux guider le travail de Bachelor et surtout le développement du projet.

Pendant le développement, il fallait également noter tous les problèmes qu'il y a eu et comment ils ont été résolu.

2. Préparation post-développement

Cette partie contient toutes les informations qui ont été analysé et développé pour préparer au mieux le développement du site internet.

2.1 THE DAO

2.1.1 Présentation

THE DAO (Decentralized Autonomous Organisation) est un fond d'investissement décentralisé et autonome, qui permet aux personnes qui y ont investi des Ethers, de voter pour des propositions de projets. Cet outil permet d'investir sur des propositions de projet qui seront développés sur la blockchain Ethereum.

Pour investir de l'argent dans THE DAO, il faut acheter des DAO Tokens à l'aide d'Ethers. Lorsqu'il y a une proposition de projet, une personne peut voter oui ou non, le oui va être incrémenté avec la balance de l'utilisateur qui a voté et le non sera incrémenté de la même manière.

Par exemple, si une personne vote oui en ayant 10 DAO Tokens et une personne vote non en ayant investi 4 DAO Tokens, le oui va l'emporter car il vaudra 10, alors que le non vaudra 4. Bien évidemment tout ceci se fera avec des milliers de personnes, donc ces sommes s'incrémenteront.

Cela permet d'avoir un pouvoir de vote, proportionnel à son investissement. Les personnes qui auront plus d'argent dans le fond d'investissement auront une plus grande répercussion sur les choix.

C'est un fonctionnement similaire à une SA, les personnes ayant le plus d'actions auront un fort pouvoir de décision, où même une personne possédant 51% des actions pourra être le seul décisionnaire.

Le but de THE DAO est de permettre d'avoir la somme et l'avis de plusieurs personnes sur des propositions de projet, pour ensuite savoir s'il mérite d'obtenir un investissement ou non. Tout ça en ayant un code qui fonctionne automatiquement et donc n'est pas géré par une tierce personne en qui les personnes doivent avoir confiance. C'est de là que vient le « A » qui signifie autonome.

2.1.2 Faille exploitée

THE DAO fonctionne avec des smarts contracts, dont le code est disponible pour tout le monde sur Git hub. C'est en analysant ce code, que le 17 juin 2016, une personne ou un groupe de personnes, ont pu repérer et exploiter une faille du code pour retirer environ 3,5 millions d'Ethers du DAO (environ 50 millions de dollars).

La faille qui a été exploitée, c'est un problème d'ordre dans le code. La fonction appelée pour transférer l'argent vers un autre compte se trouve avant celle où la balance du compte est mise à 0, du coup en faisant un appel récursif, le ou les assaillants ont réussi à récupérer sur un DAO child les 3,5 millions d'Ethers.

2.1.3 Analyse du code défaillant

Dans le Git hub, on peut accéder au code à différentes périodes, donc ce bout de code a été récupéré à peu près au moment où la faille a été exploitée :

```
683
684         // Burn DAO Tokens
685         Transfer(msg.sender, 0, balances[msg.sender]);
686         withdrawRewardFor(msg.sender); // be nice, and get his rewards
687         totalSupply -= balances[msg.sender];
688         balances[msg.sender] = 0;
689         paidOut[msg.sender] = 0;
690         return true;
691     }
692
```

https://github.com/slockit/DAO/blob/8d24443909191842d94e343ef6f734046ae4bc24/DAO.sol

De la ligne 683 à la ligne 691

Dans l'image ci-dessus, on voit le bout de code qui contient la faille de THE DAO. On peut voir que la fonction « Transfert » est appelée avec en paramètre la balance de la personne qui enclenche ce code. A l'intérieur de la fonction, une ligne vérifie que la balance n'est pas à 0 avant d'enclencher la transaction, le problème est qu'une transaction peut prendre plusieurs minutes et que la balance de la personne est mise à 0 uniquement après que la transaction ait été effectuée. Donc, enclenchant plusieurs fois d'affilé le code ci-dessus, la fonction « Transfert » est lancé avec succès à chaque fois, tant que la première qui a été lancée n'a pas terminé la transaction et mis la balance à 0.

Voir cette erreur permet de prendre conscience qu'il faut faire extrêmement attention à l'ordre de nos fonctions et opérations. Il faut également bien tester l'application avant de la mettre à disposition, surtout quand elle gère de l'argent, pour éviter un maximum de bugs ou failles. Il y a tellement de possibilités dans une application, qu'il peut y avoir quand même des bugs même après avoir testé l'application, donc il faut se préparer à cette éventualité pour pouvoir réagir assez rapidement, lorsqu'il y a un problème. On peut faire en sorte de minimiser les failles qu'il y a sur les fonctions concernant l'argent, en prenant le risque d'avoir plus de problèmes sur d'autres fonctions. Il faut que les tests soit en priorité pour les parties de l'application où il y a de l'argent en jeu.

2.1.4 Mesures prises

Le temps pour pouvoir récupérer les ethers du DAO child est d'environ 27 jours, ce qui a permis de réfléchir à une solution après l'exploitation de la faille. Un vote a été proposé pour choisir une solution. Tout d'abord il y a eu un gel de tous les ethers (ceux des assaillants et ceux des investisseurs) de THE DAO, puis la solution d'un hard fork pour faire une modification dans Ethereum pour annuler l'exploitation de la faille, a été proposée. Les mineurs ont été partagés, plusieurs d'entre eux étaient d'accord pour cette idée, tandis que d'autres disait que c'était contraire à une des idées fondamentales de la blockchain : L'immuabilité.

Ce qui s'est passé, c'est qu'une copie de la blockchain Ethereum a été faite avec la modification pour éviter la faille, tout en gardant la blockchain de base pour les personnes qui étaient contre. La blockchain de base a été appelée Ethereum Classic, ce qui fait que les personnes qui avaient des ethers avant cette attaque, ont désormais également des ethers classic (ETC).

2.1.5 Analyse du code de la fonction vote

La fonction de vote du projet THE DAO a été analysée, pour voir comment les votes sont gérés et comptabilisés, pour apporter une idée de scénario pour le site internet :

```
486
487
488     function vote(uint _proposalID, bool _supportsProposal) onlyTokenholders noEther {
489
490 1.      Proposal p = proposals[_proposalID];
491         if (p.votedYes[msg.sender]
492             || p.votedNo[msg.sender]
493             || now >= p.votingDeadline) {
494
495             throw;
496         }
497
498 2.      if (_supportsProposal) {
499             p.yea += balances[msg.sender];
500             p.votedYes[msg.sender] = true;
501         } else {
502             p.nay += balances[msg.sender];
503             p.votedNo[msg.sender] = true;
504         }
505
506 3.      if (blocked[msg.sender] == 0) {
507             blocked[msg.sender] = _proposalID;
508         } else if (p.votingDeadline > proposals[blocked[msg.sender]].votingDeadline) {
509             // this proposal's voting deadline is further into the future than
510             // the proposal that blocks the sender so make it the blocker
511             blocked[msg.sender] = _proposalID;
512         }
513
514         Voted(_proposalID, _supportsProposal, msg.sender);
515     }
```

https://github.com/slockit/DAO/blob/8d24443909191842d94e343ef6f734046ae4bc24/DAO.sol

De la ligne 488 à la ligne 515

La fonction vote prend en paramètre l'identifiant de la proposition (*_proposalID*) auquel on veut voter et un *boolean* (*_supportsProposal*) qui est à *true* si la personne est pour la proposition et *false* si la personne est contre la proposition.

2.1.5.1 Bloc de code numéro 1

La variable *p* de type *Proposal* est initialisée avec la proposition en question qui est récupérée dans le tableau de proposition (*proposals[]*) grâce à son

identifiant(*_proposalID*).
Ensuite une vérification est faite, permettant de savoir si la personne a déjà voté pour cette proposition ou si la date et l'heure de fin de votation est déjà dépassée. La vérification se fait en cherchant si à la position du *msg.sender* dans le tableau « oui » ou le tableau « non » (*p.votedYes[msg.sender]* || *p.votedNo[msg.sender]*), il y a *true*. Si aucun des deux ne renvoie *true*, ça vérifie que la date et l'heure du jour (*now*) est plus grande que la date et l'heure de la deadline (*p.votingDeadline*). Lorsqu'une des trois conditions renvoi un *true*, le contenu du *if* est exécuté et donc arrête la fonction (*throw*), car soit la personne a déjà voté, soit la date de fin de vote pour la proposition est dépassée.

2.1.5.2 Bloc de code numéro 2

Le paramètre (*_supportsProposal*) est testé, si il est égal à *true*, alors la personne est pour la proposition et donc on incrémente le paramètre *yea* avec la balance du votant (*yea* += *balances[msg.sender]*) et on rend sa position *true* dans le tableau des « oui » de la proposition (*p.votedYes[msg.sender]* = *true*).
Si le paramètre (*_supportsProposal*) est égal à *false*, alors on se retrouve dans le *else* et on fait la même chose mais avec le tableau des « non » (*p.votedNo[msg.sender]* = *true*) et le paramètre *nay* est incrémenté avec la balance du votant (*p.nay* += *balances[msg.sender]*).

2.1.5.3 Bloc de code numéro 3

Le tableau blocked contient à la position du votant l'identifiant de la proposition dont la deadline est la plus grande, cela permet d'empêcher les personnes participant à la votation d'une proposition, de retirer ses "DAO tokens" pendant la période de vote. Une vérification est faite dans le tableau *blocked* à la position du votant et ça regarde si le nombre contenu est égal à 0 (*blocked[msg.sender]==0*), s'il est égal à 0 alors cela signifie que la personne n'a pas voté pour une proposition en cours et donc le numéro attribué à la case du tableau est le numéro de l'ID de la proposition dans laquelle il vient de voter (*blocked[msg.sender=_proposalID)*.
Si le contenu de blocked ne contient pas 0, alors cela signifie que la personne a déjà voté sur une autre proposition, donc les deadline des 2 propositions sont comparées et si celle actuelle se termine plus tard que celle contenu (*p.votingDeadline>proposals[blocked[msg.sender]]*), alors l'identifiant sera remplacé par celui de la proposition en cours (*blocked[msg.sender] = _proposalID*).

2.2 Stellar

Stellar est une plateforme qui utilise la technologie blockchain pour effectuer des transactions entre divers comptes. Sa crypto-monnaie est les Lumens. Pour avoir les différents états d'un comptes, il y a plusieurs serveurs contenant les dernières informations et ensuite avant de mettre à jour ces informations, tous les serveurs font un consensus, c'est-à-dire il faut que plus de 50% des serveurs aient les même informations avant d'accepter un changement pour un compte. C'est ça la technologie blockchain, ce qui permet de ne pas altérer les informations.

Stellar fournis des services à travers son API : Horizon, qui permet de développer des applications pour faire des paiements et transfère d'argent passant par son network. On peut également créer sa propre monnaie sur son site pour l'utiliser pour faire des votes, de la réputation, des points de fidélités, etc.

Plusieurs applications utilisent Stellar pour faire des transferts d'argent entre plusieurs pays étrangers.

Une des choses intéressante est qu'en possédant un compte Stellar, on peut l'utiliser dans toutes les applications qui interagissent avec Horizon.

Stellar est bien documenté pour les développeurs et fournis même une API qui permet de faire des tests avec de la fausse monnaie, sur un différent network, ce qui permet de ne pas polluer le network public. L'API de test s'appelle Horizon Testnet.

De plus, pour éviter que beaucoup de comptes soient créés et gérés pour rien, il faut avoir un minimum de 20 Lumens avant que le compte soit actif sur Horizon public.

2.3 Cowaboo

Cowaboo est une plateforme permettant de publier des informations sur divers sujets en créant un observatory pour un thème global. Ensuite dans chaque observatory, on peut faire des entry qui sont différentiées et nommées à l'aide de tag. On peut choisir qui est dans l'observatory et comment on accepte les changements d'entry. On peut soit proposer un changement et le valider individuellement, soit demander à ce qu'il y ait un consensus entre tous les membres du projet, pour que le changement soit accepté et effectué.

Cette plateforme est utile pour les collaborateurs entre eux, les professeurs voulant partager des informations avec leurs élèves et les élèves entre eux pour partager les informations qu'ils trouvent sur des sujets qui peuvent intéresser les autres.

De plus, Cowaboo possède une API qui permet de poster, récupérer, modifier des observatory, entry. Ce qui permet de pouvoir faire des post depuis une autre application, mais qui permet également de l'utiliser en tant que « base de donnée ». On poste des entry contenant nos donnée à travers l'API et on peut y accéder depuis n'importe où, du moment qu'il y a une connexion internet.

Cowaboo utilise Stellar, il fournit des comptes lors de l'inscription et utilise une monnaie créée appelée Energy. Cette monnaie permet de voter pour des entry d'autres participants lorsqu'on les apprécie. Les Energy est un token, qui permet de descendre en dessous de 0 et donc d'être en négatif.

3. Liste des scénarios

Il y a plusieurs scénarios possibles découlant de l'idée de base du projet. Plusieurs scénarios vont être listés, pour pouvoir les comparer et savoir lesquels sont les plus efficaces et les plus réalisables. On regarde pour chaque scénario ces avantages et inconvénients, ce qui permet de les comparer plus facilement et d'éviter plusieurs problèmes, qui n'auraient pas été soulevés dans le cas où le projet aurait été attaqué directement avec un seul scénario.
De plus, dans le cas où une des possibilités du scénario, ne peut pas être implémentée, on peut utiliser à la place, une possibilité d'un des autres scénarios.

3.1 Scénario 1

Description du scénario :

Chaque projet proposé est associé à un compte et les personnes qui investissent pour le projet envoient directement leur crypto-monnaie à l'adresse du publique du compte du projet sélectionné.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : Aucune.
Flow de base :

1. L'utilisateur clique sur le bouton investir d'un projet.
2. Le système affiche la clé publique du projet.
3. L'utilisateur va sur son wallet pour faire la transaction en utilisant la clé publique fournit.
4. Le système met à jour le total de l'investissement du projet en question.

Avantages du scénario :

- Le projet est plus facilement réalisable donc moins de risque d'échec du projet.
- Les utilisateurs peuvent utiliser leur wallet qu'ils possèdent déjà pour faire la transaction.
- Pas de possibilité de bug ou de faille avec l'argent, car le site fournit que des adresses publiques, mais ne gère pas lui-même les crypto-monnaies.
- Permet aux personnes qui proposent des projets de définir quel type de crypto-monnaie elles veulent récupérer.
- Les personnes voulant investir n'ont pas besoin de s'inscrire/se connecter au site pour faire leur investissement.
- Les utilisateurs peuvent voter pour plusieurs projets.

Inconvénients du scénario :

- Le site a juste l'utilité de proposer les projets mais ne fait rien de plus.
- Les personnes voient les projets sur le site, mais lorsqu'elles veulent investir sur un projet, elles doivent aller sur un autre site / une autre application pour faire la transaction.
- La personne proposant le projet, peut partir avec l'argent directement, car c'est son compte qui contient les investissements.
- On ne peut pas faire en sorte que seul le gagnant récupère l'investissement, car le site ne gère pas les comptes des proposants et ne peut pas les forcer à rendre l'argent lorsque leur projet ne gagne pas.

3.2 Scénario 2

Description du scénario :

Chaque compte utilisateur possède une possibilité de 2 votes, il peut utiliser ses 2 votes pour un projet ou bien partager ses 2 votes pour 2 projets. Lorsqu'un projet est vainqueur, les personnes ayant voté pour le projet envoient leur investissement au compte du projet gagnant.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet.
Flow de base :

1. L'utilisateur clique sur le bouton de vote d'un projet.
2. Le système prend en compte le vote et le décrémente de l'utilisateur.
3. L'utilisateur clique sur le bouton de vote d'un second projet (ou du même).
4. Le système prend en compte le vote et le décrémente de l'utilisateur.
5. Lorsqu'un projet est gagnant le système envoie l'adresse du compte gagnant aux utilisateurs ayant voté pour le projet.

Avantages du scénario :

- Le projet est plus facilement réalisable donc moins de risque d'échec du projet.
- Les utilisateurs peuvent utiliser leur wallet qu'ils possèdent déjà pour faire la transaction.
- Pas de possibilité de bug ou de faille avec l'argent, car le site fournit que des adresses publiques, mais ne gère pas lui-même les crypto-monnaies. Les possibles failles peuvent être effectués avec les votes, c'est moins grave que de directement détourner de l'argent.
- Permet aux personnes qui proposent des projets de définir quel type de crypto-monnaie elles veulent récupérer.
- Les utilisateurs peuvent voter pour plusieurs projets.

Inconvénients du scénario :

- Les personnes voient les projets sur le site, mais lorsqu'elles veulent investir sur un projet, elles doivent aller sur un autre site / une autre application pour faire la transaction.
- On ne peut pas être sûre que toutes les personnes ayant voté, investissent de l'argent dans le projet. Donc, il se peut qu'un projet ayant eu moins de votes que le gagnant, aurait eus plus d'investissements.
- Les votants ne regardent pas autant en détail le projet pour lequel ils votent que lorsqu'ils misent directement leur argent. Ce qui fait que lors de la demande d'investissement, les personnes regardent plus en détail le projet et voient quelque chose qui ne leur plait pas, mais qu'ils n'avaient pas vu leur du vote et donc décident de ne pas investir, alors qu'ils ont permis au projet de gagner grâce à leur vote.

3.3 Scénario 3

Description du scénario :

Chaque compte utilisateur possède des tokens créés pour le site. Ils doivent acheter les tokens avec une certaine crypto-monnaie. Lorsqu'ils votent pour un projet, leur balance est comptabilisée pour le projet et est bloquée jusqu'à la fin du temps indiqué sur le projet. Si le projet auquel la personne investit atteint son objectif, la balance de la personne est débitée de son compte et est crédité sur le compte du proposant. Les personnes ayant voté pour un projet qui n'a pas atteint son but, ont juste leur balance débloquée et peuvent revoter pour un autre projet. Les personnes peuvent ré-échanger leurs tokens avec la crypto-monnaie avec laquelle ils les ont achetés.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet et posséder des tokens.

Flow de base :

1. L'utilisateur clique sur le bouton de vote d'un projet.
2. Le système prend en compte la balance de l'utilisateur et la bloque jusqu'à la fin du vote pour ce projet.
3. Lorsque la fin du vote du projet est atteinte, la balance des votants est créditée sur le compte du proposant.

Flow altérnatif :

3. Le projet n'a pas atteint son but, le système débloque la balance des votants.

Avantages du scénario :

- Tout le monde vote avec la même monnaie : les tokens du site.
- Ce scénario s'inspire beaucoup de THE DAO, donc on peut déjà savoir à quel genre de problème on peut se préparer et avoir déjà des informations sur le fonctionnement du projet.
- Avoir ses propres tokens pour le site donne une identité au site et rassemble une communauté.

Inconvénients du scénario :

- Il peut y avoir des failles dans le code et donc des personnes malveillantes peuvent les exploiter pour récupérer l'argent contenu sur le site internet.
- Le cours des monnaies change, donc il faut gérer ceci lors des achats et ventes de tokens.
- Les personnes peuvent voter uniquement sur un projet à la fois.
- On ne peut pas définir de vainqueur pour un éventuel investissement / soutient de la HEG, car les projets ont leur propre temps de vote et ne sont pas en concurrence.

3.4 Scénario 4

Description du scénario :

Chaque compte utilisateur possède des tokens créés pour le site. Ils doivent acheter les tokens avec une certaine crypto-monnaie. Lorsque les personnes votent pour un projet, elles peuvent choisir combien de tokens elles souhaitent mettre pour un projet en particulier. Ces tokens sont gelé dans leur balance. Avec les tokens restant elles peuvent voter pour d'autres projets. Lorsqu'un projet a atteint son objectif, les tokens sont crédités sur le compte du proposant. Si le projet n'a pas atteint l'objectif, les tokens des votants sont dégelés. Les personnes peuvent rechanger leurs tokens avec la crypto-monnaie avec laquelle ils les ont achetés.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet et posséder des tokens.

Flow de base :

1. L'utilisateur rentre le nombre de tokens qu'il souhaite investir dans le projet et ensuite clique sur le bouton investir.
2. Le système prend en compte les tokens de l'utilisateur et les bloques jusqu'à la fin du vote pour ce projet.
3. Lorsque la fin du vote du projet est atteinte, les tokens des votants sont crédités sur le compte du proposant.

Flow altérnatif :

3. Le projet n'a pas atteint son but, le système débloque les tokens des votants.

Avantages du scénario :

- Tout le monde vote avec la même monnaie : les tokens du site.
- Ce scénario s'inspire beaucoup de THE DAO, donc on peut déjà savoir à quel genre de problème on peut se préparer et avoir déjà des informations sur le fonctionnement du projet.
- Avoir ses propres tokens pour le site donne une identité au site et rassemble une communauté.
- Les personnes peuvent voter / investir pour plusieurs projets.

Inconvénients du scénario :

- Il peut y avoir des failles dans le code et donc des personnes malveillantes peuvent les exploiter pour récupérer l'argent contenu sur le site internet.
- Le cours des monnaies change, donc il faut gérer ceci lors des achats et ventes de tokens.

- On ne peut pas définir de vainqueur pour un éventuel investissement / soutient de la HEG, car les projets ont leur propre temps de vote et ne sont pas en concurrence.

3.5 Scénario 5

Description du scénario :

Chaque utilisateur à un compte pour le site. Lors de leur inscription, les clés privés et publics leurs sont fournies pour qu'ils puissent remplir leur compte avec certaines crypto-monnaies. Lorsqu'ils souhaitent voter pour un projet, ils choisissent combien de leur monnaie ils veulent investir dans un projet et la monnaie est transférée sur le compte du proposant.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet et avoir rempli son compte.

Flow de base :

1. L'utilisateur choisi la monnaie et rentre la valeur qu'il souhaite investir dans le projet et ensuite clique sur le bouton investir.
2. Le système transfert la monnaie vers le compte du proposant.

Avantages du scénario :

- Les personnes peuvent voter / investir pour plusieurs projets.
- L'investissement est fait en même temps que le vote, donc il y a moins d'étapes.

Inconvénients du scénario :

- Lorsqu'une personne investie pour un projet, elle envoie la monnaie et ne la récupère plus.
- Les proposants peuvent envoyer l'argent de l'investissement sur un autre compte avant la fin des votes.
- Les personnes osent moins voter sachant que dans tous les cas l'argent voté ne revient pas.

3.6 Scénario 6

Description du scénario :

Chaque utilisateur à un compte pour le site. Lors de leur inscription, les clés privés et publics leurs sont fournies pour qu'ils puissent remplir leur compte avec certaines crypto-monnaies. Lorsqu'ils souhaitent voter pour un projet, ils choisissent combien de leur monnaie ils veulent y investir et la monnaie est transférée sur le compte de gestion du site internet. Lorsque le vainqueur est défini, l'investissement pour son projet qui est présent sur le compte de gestion est transféré sur le compte du projet, moins les taxes.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet et avoir rempli son compte.

Flow de base :

1. L'utilisateur choisi la monnaie et rentre la valeur qu'il souhaite investir dans le projet et ensuite clique sur le bouton investir.
2. Le système transfert la monnaie vers le compte de gestion et prend en compte la somme pour le projet.
3. Lorsqu'un projet est vainqueur, le compte de gestion transfère la monnaie correspondante aux votes / investissements du projet, moins les taxes.

Flow alternatif :

3. Lorsqu'un projet est perdant, le compte de gestion transfère la monnaie correspondante investissements des personnes, moins les taxes, à chacun d'entre eux.

Avantages du scénario :

- Les personnes peuvent investir pour plusieurs projets.
- Le site gère la monnaie entre les votes, donc la monnaie est transférée au vainqueur, lorsque son projet a gagné.
- Si un projet ne gagne pas, la monnaie est redistribuée au votant, donc ils oseront plus voter.

Inconvénients du scénario :

- Gérer la création de compte et gérer les transactions est quelque chose de nouveau pour le développeur, donc risque d'échec du projet.
- Lorsqu'un projet n'est pas choisi les votants récupèrent un peu moins d'argent qu'ils n'avaient, car il y a des taxes lors des transactions.
- Il n'y a que le vainqueur qui récupère l'investissement, donc il peut y avoir des bons projets qui ont eu juste un peu moins d'investissement, mais qui ne récupèrera pas l'argent.

3.7 Scénario 7 (Choisi)

Description du scénario :

Chaque utilisateur à un compte pour le site. Lors de leur inscription, les clés privés et publics leurs sont fournies pour qu'ils puissent remplir leur compte avec certaines crypto-monnaies. Lorsqu'ils souhaitent voter pour un projet, ils choisissent combien de leur monnaie ils veulent y investir et la monnaie est transférée sur le compte de gestion du site internet. Lorsqu'un projet atteint un certain objectif, il récupère l'investissement à la fin du vote. Le vainqueur récupère l'investissement des votants et gagne une possibilité de soutien supplémentaire, de la part de la HEG. Puisqu'il faut atteindre un certain objectif pour recevoir l'investissement, un projet n'ayant pas atteint l'objectif ne peut pas être le vainqueur. Donc il se peut que lors d'une période de vote, il n'y ait aucun vainqueur.

Acteurs : Utilisateur « investisseur ».
Déclencheur : L'utilisateur souhaite investir pour un projet proposé.
Précondition : L'utilisateur doit être connecté sur le site internet et avoir rempli son compte.

Flow de base :

1. L'utilisateur choisi la monnaie et rentre la valeur qu'il souhaite investir dans le projet et ensuite clique sur le bouton investir.
2. Le système transfert la monnaie vers le compte de gestion et prend en compte la somme pour le projet.
3. Lorsqu'un projet atteint l'objectif, le compte de gestion transfère la monnaie correspondante aux investissements du projet, moins les taxes.

Flow alternatif :

3. Lorsqu'un projet n'atteint pas l'objectif, le compte de gestion transfère la monnaie correspondante aux investissements des personnes, moins les taxes, à chacun d'entre eux.

Avantages du scénario :

- Les personnes peuvent voter / investir pour plusieurs projets.
- Le site gère la monnaie entre les votes, donc la monnaie est transférée au vainqueur, lorsque son projet a gagné.
- Si un projet ne gagne pas, la monnaie est redistribuée aux votants, donc ils oseront plus voter.
- Plusieurs projets peuvent obtenir un investissement, pas seulement le vainqueur.

Inconvénients du scénario :

- Gérer la création de compte et gérer les transactions est quelque chose de nouveau pour le développeur, donc risque d'échec du projet.

- Lorsqu'un projet n'est pas choisi, les investisseurs récupèrent un peu moins d'argent qu'ils n'avaient, car il y a des taxes lors des transactions.

3.8 Choix du scénario

Les différents scénarios découlent de discussions avec le professeur de suivi du projet. On a développé plusieurs idées et au fur et à mesures des discussions, plusieurs éléments s'ajoutaient ou se modifiaient.

Le choix final du scénario à développer dans le site internet est le scénario 7. Pour arriver au choix de ce scénario, plusieurs problèmes ont été évoqués. Tout d'abord, le problème est qu'en faisant uniquement des votes pour les personnes qui trouvent un projet intéressant, on ne peut être certain, que toute les personnes qui ont voté, vont investir pour ce projet. C'est pourquoi, on a décidé de faire en sorte que les votes soient faits en investissant directement l'argent.

Ensuite, Il y a le problème qu'en faisant un investissement directement sur le compte du proposant, il peut récupérer l'argent directement et ne le rendra pas aux votants, si son projet n'a pas été choisi. Donc, on a décidé de faire un compte de gestion, qui permet de verser les investissements aux gagnants et de rendre le reste aux personnes ayant investi sur un projet non sélectionné.

3.9 Défis techniques

Le développement de ce scénario va apporter des défis techniques, donc chacun est expliqué ici. Ces défis techniques ont été définis avant le début du développement.

3.9.1 Gestion d'un compte Stellar

Il va falloir savoir gérer un compte directement par code concernant le compte de gestion appartenant au site. Le site devra être au courant de toutes les transactions faites vers son compte et devra savoir quoi en faire et à quel moment.

3.9.2 Gestion des transactions

Il y aura aussi la gestion des transactions des investisseurs, il faudra voir comment réussir une transaction, comment savoir qu'une transaction a été faite et ensuite retenir quelles transactions sont pour quels projets, pour pouvoir les incrémenter. Il faut également pouvoir récupérer l'id des transactions, pour les afficher dans le moniteur d'un projet, pour savoir quelles sommes ont été investies.

3.9.3 Gestion des taxes

La gestion des taxes fera aussi parti d'une réflexion, il faudra savoir comment payer les taxes de transaction, de combien sont-elles et donc, comment les calculer pour que la transaction soit valide, sachant qu’il faut que le compte reste au-dessus de 20 Lumens.

3.9.4 Gestion du temps

Les transactions prennent parfois du temps à se faire, c'est d'ailleurs à cause du temps de réalisation d'une fonction que THE DAO a eu une faille qui a été exploitée. Il faut développer en vérifiant que s'il y a un temps d'une fonctionnalité trop long, il faut en prendre compte et trouver une solution, pour que l'utilisateur ne soit pas bloqué longtemps ou pour qu'il ne voit pas le temps que met l'opération. Il faut également gérer le lancement de plusieurs fonctions, sachant que certaines ont besoin du résultat d’autres pour s’exécuter correctement.

3.9.5 Gestion de la sécurité

Il ne faut pas que les clés privées soient enregistrées, l'utilisateur doit rentrer la clé privée uniquement pour se connecter et faire des transactions, la clé publique peut être enregistrée par contre.

4. Présenation d’InvestProject

La présentation du site permet de savoir que fait le site et comment il fonctionne de manière générale. Le détail sur les chiffres et les règles précises se trouve dans la partie règlement d’InvestProject.

4.1 Présentation

InvestProject est un site internet permettant de proposer des projets pour obtenir un investissement. Il y a pour chaque périodes des propositions créées par des utilisateurs et ceux qui souhaitent investir pour un des projets, parcourent et choisissent la/les propositions qui les a/ont le plus séduit.

Le but de cela est d’avoir un site ou les personnes souhaitant investir sur des projets et les personnes souhaitant obtenir des investissements, se rencontrent et ont directement plusieurs choix de projets à disposition au même endroit.

4.1.1 Inscription

Les auteurs des projets doivent s'inscrire sur le site pour pouvoir y publier une proposition. Les investisseurs qui vont mettre de l'argent sur des propositions doivent également être inscrits sur le site.

Lorsque les utilisateurs sont inscrits, ils reçoivent par mail leur clé privée et leur clé publique, qui vont permettre de faire ou de recevoir les investissements.

4.1.2 Investissements / Transactions

Les transactions entre les utilisateurs et les propositions se font en passant par l’api Horizon qui interagit avec la blockchain de Stellar. La crypto-monnaie utilisée pour faire les investissements est : les Lumens.

Tous les investissements sont transférés sur le compte de gestion d’InvestProject, qui va ensuite les distribuer aux bons comptes, à la fin de la période.

Lors de l’inscription, le compte créé pour les utilisateurs est logiquement à 0, il faut donc suivre le tutoriel qui explique comment recharger ce compte avec des Lumens.

4.1.3 Périodes

Il y a des périodes d’un mois pour que les personnes fassent des propositions ou/et investissent sur des projets. A la fin de chaque période, il y a la distribution des investissements et une nouvelle période est lancée.

La distribution des investissements se fait en fonction du nombre de total de Lumens récupéré par chaque proposition. Il y a un objectif commun d'argent à récupérer pour que l'investissement soit validé.

Pour avoir une bonne vision, l'objectif est défini en CHF. Donc, chaque champ contenant un nombre de Lumens, affiche son équivalent en CHF. Il y a également le prix d'un Lumens affiché sur le site.

4.1.4 Distribution des Lumens

Lorsqu'un projet atteint l'objectif, le compte d'InvestProject lui envoie les Lumens correspondant à l'investissement récupéré. Par contre, lorsqu'un projet n'atteint pas l'objectif, les Lumens sont redistribués aux comptes des utilisateurs qui ont investi sur le projet.

Exemple du déroulement d'une fin de période d'investissement avec l'objectif commun de 3000 CHF.

Liste des propositions de projet :

Projet A : Faire un jeu pour smartphone 900 CHF

Projet B : Réalisation d'une application pour les places de parkings 2000 CHF

Projet C : Réalisation d'un forum pour les élèves des Hautes Ecoles 3200 CHF

Projet D : Création d'un site de vente d'accessoires de smartphones 0 CHF

Investissement versé :

Le projet « C » est le seul projet à avoir atteint l'objectif des 3000 CHF, il sera donc le seul à récupérer l'investissement de 3200 CHf.

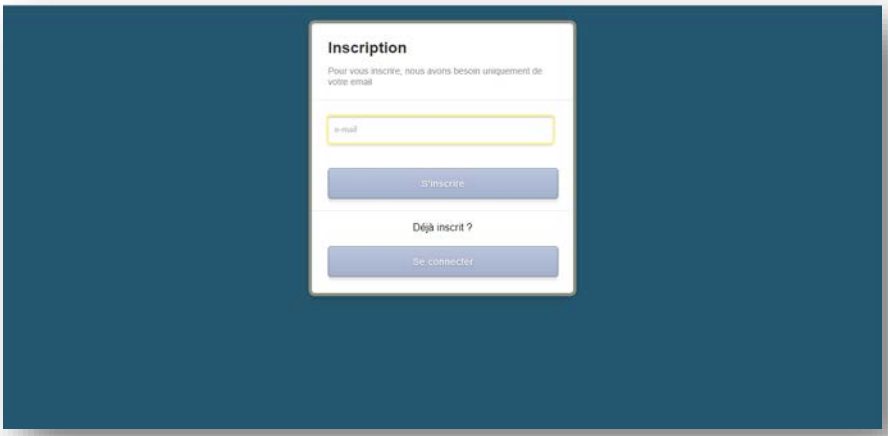
Investissement rendu :

Les projets « A » et « B » verront l'argent investit retourner vers les investisseurs, car ils n'ont pas atteint l'objectif des 3000 CHF minimum.

4.2 Description du contenu d'InvestProject :

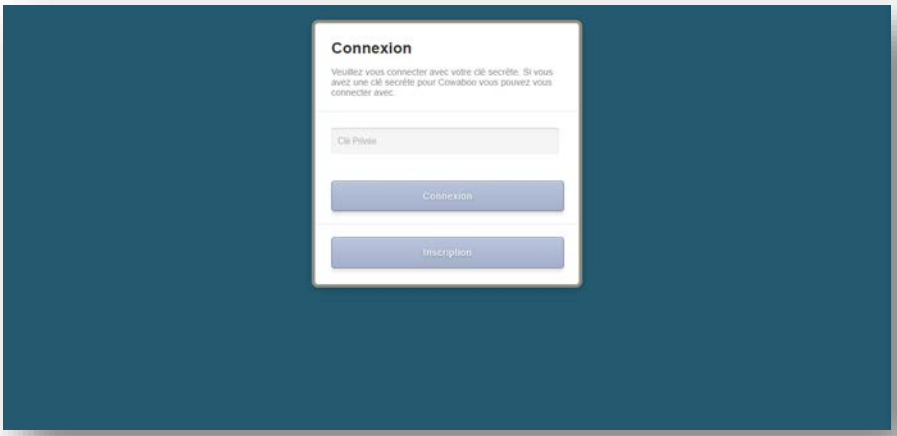
Ici, toute les pages d'InvestProject sont décrites, une à une, associées du printscreen de la page.

4.2.1 Inscription



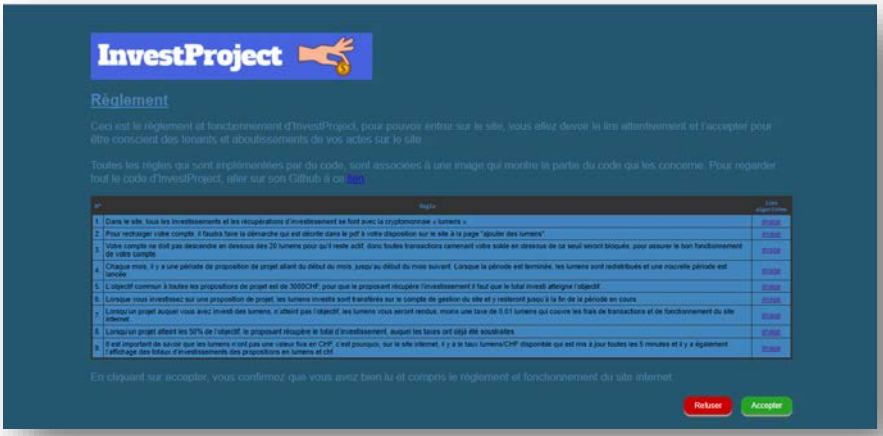
La page « inscription » permet aux personnes de créer un compte en entrant leur adresse email. Suite à ça, elles vont recevoir leur clé publique et privée par email.

4.2.2 Connexion



La page « connexion » permet de se connecter au site en utilisant sa clé privée.

4.2.3 Acceptation

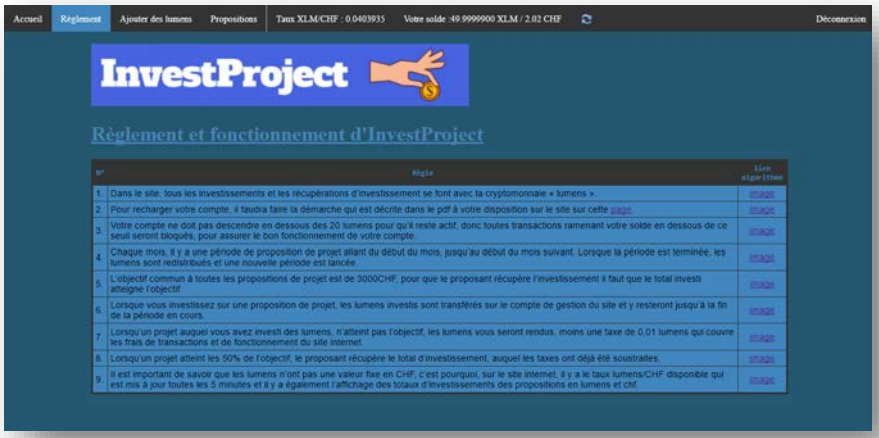


Cette page s'affiche lors de la première connexion de l'utilisateur. Elle affiche le règlement / fonctionnement d'InvestProject et pour entrer dans le site, l'utilisateur doit accepter ce règlement en cliquant sur le boutton « accepter ». Tant que le règlement n'est pas accepté, à chaque connexion cette page s'affiche.

4.2.4 Accueil

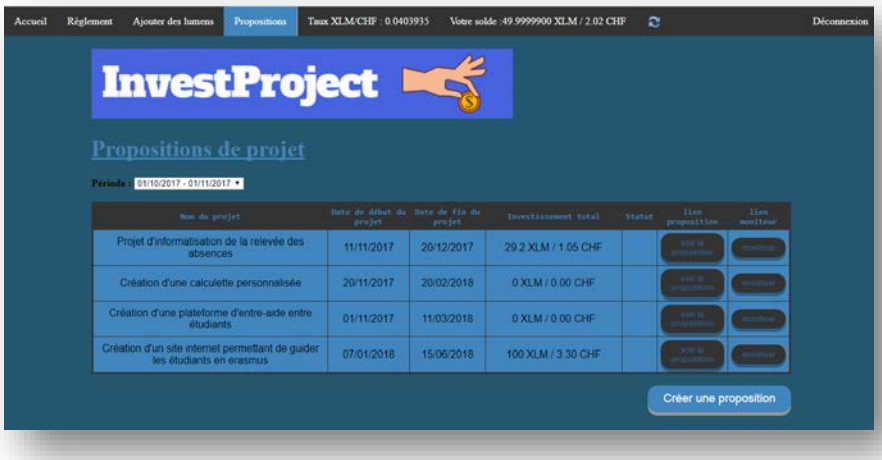
Dans la page « accueil » se trouve une explication du principe et du but du site internet, pour que les visiteurs sachent où ils se trouvent.

4.2.5 Règlement



Dans la page « règlement », il y a la liste des règles pour que les utilisateurs soit au courant de ce qui se passe sur le site avec leurs Lumens et de son fonctionnement. Cette page est là pour qu'ils puissent avoir accès aux règles quand ils le souhaitent pour avoir un rappel, mais ce règlement doit avoir été accepté lors de la première connexion.

4.2.6 Liste des propositions

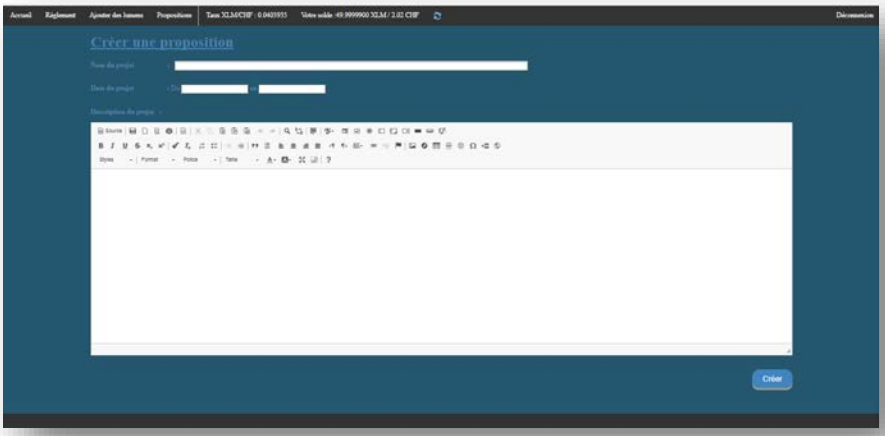


Dans la page « liste des propositions », il y a la liste des propositions de projets pour la période en cours, avec pour chaque proposition, son nom, la date de début et de fin du projet, le nombre de Lumens / CHF reçus, le lien pour aller voir la proposition en détail et le lien du moniteur contenant un historique des investissements.

De plus, il y a une liste déroulante avec les différentes périodes, permettant de voir les propositions des périodes précédantes.

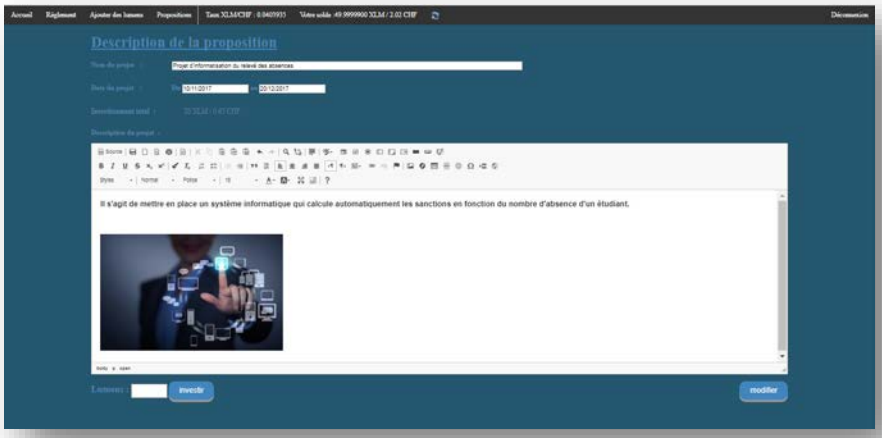
Il y a également un bouton qui permet d'aller sur la page de création d'une proposition.

4.2.7 Création de proposition



Cette page est la page qui permet de créer sa proposition, elle contient les champs de dates qui sont remplis en sélectionnant des dates sur les calendriers qui apparaissent lorsqu'on clique sur les champs. De plus, il y a un éditeur de texte qui permet de mettre en forme et soigner sa présentation de la description du projet, et d'y ajouter par exemple des images.

4.2.8 Proposition



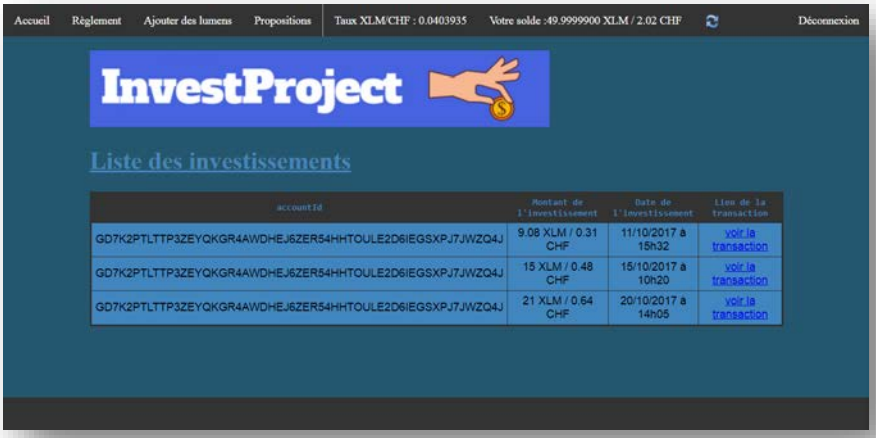
Dans cette page il y a le détail d'une proposition de projet en particulier, avec son nom, la somme actuelle investie, la date de début et de fin de projet, ainsi qu'une description du projet pour l'expliquer et convaincre les personnes de voter pour ce projet. Les

personnes ayant un site ou un fichier contenant plus de détail pour le projet, pourront mettre le lien de leur site ou associer leur fichier dans la description.

Pour ceux qui souhaitent investir, il y a un champ où il faut mettre le nombre de Lumens pour l'investissement. La possibilité d'investissement se trouve uniquement dans le détail d'une proposition, pour éviter les erreurs d'investissement et donc que les utilisateurs soit sur d'investir pour le bon projet.

Pour les auteurs de la proposition, il y a le bouton « modifier » qui est actif, qui permet d'enregistrer les modifications faites dans les champs. Les champs et le bouton sont évidemment inactifs pour les autres utilisateurs.

4.2.9 Moniteur



Cette page contient l'historique des investissements faits pour la proposition de projet correspondante. Chaque ligne, contient la clé publique de l'investisseur, la date et l'heure de l'investissement, le nombre de Lumens investis et le lien vers une représentation en JSON de la transaction complète.

5. Problèmes et solutions

Durant le développement d'InvestProject, il y a eu plusieurs problèmes auquel du temps y a été accordé pour trouver une solution. Tous ces problèmes sont listés et pour ceux dont la solution a été apportée avec du code, il y aura un printscreen du code en question. Ici sera affiché les bouts de codes qui concernent certains problème, pour trouver le code en intégralité, il est disponible sur GitHub à ce lien : https://github.com/saidiamir/InvestProject2 .

5.1 Enregistrer les périodes et leurs propositions

5.1.1 Problème

Il faut enregistrer toutes les périodes et propositions, sans utiliser un logiciel de Base de données. Il faut que ces classes soient enregistrées sur Cowaboo grâce à son API, en les postant dans une entry.

5.1.2 Solution

Créer un tableau qui contient les périodes qui elles-mêmes contiennent un tableau avec les propositions qui les concernent. Ensuite, convertir toutes les classes contenues dans le tableau en JSON, pour ensuite envoyer le JSON en tant que value dans la requête de l'API de Cowaboo qui l'enregistre dans le fichier.

Lors de la récupération des données, faire l'inverse, c'est-à-dire récupérer le JSON et le convertir en tableau d'objets, pour pouvoir les utiliser dans le site.

5.1.3 Code

```
function majPeriodes(json, hash){
    var http=new XMLHttpRequest();
    var url = "http://groups.cowaboo.net/group-coInvest/public/api/entry?";
    var params = "secretKey="+secretInvestProject;
    params    += "&observatoryId=Propositions";
    params    += "&hash="+hash;
    params    += "&newValue="+json;
    http.open("PUT", url, true);

    //Send the proper header information along with the request+
    http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    http.setRequestHeader("Accept", "application/json");
    http.setRequestHeader("Accept", "application/x-www-form-urlencoded");
    http.send(params);
    http.onreadystatechange = function() {//Call a function when the state changes.

        if(http.readyState == 4 && http.status == 200) {
            res.send("success");
        } else if(http.readyState == 4){
            res.send("error");
        }
    }
}
```

5.2 Affichage du taux de XLM / CHF

5.2.1 Problème

Etant donné qu'un Lumens ne vaut pas 1 franc suisse, les utilisateurs ont dû mal à savoir combien ils possèdent d'argent, combien ils investissent et à combien de francs d'investissement est arrivé leur projet.

5.2.2 Solution

Afficher constamment la valeur d'un Lumens en francs Suisses. Afficher chaque valeur en Lumens et en francs Suisses.

Les XLM changeant régulièrement de valeur en CHF, il faut mettre à jour régulièrement le taux XLM/CHF. Donc, l'appelle à 2 API a été fait. La 1ère API est : http://api.fixer.io/latest qui renvoi une liste des prix où on récupère la valeur du taux euro / franc suisse, car le taux Lumens / CHF n'est pas disponible. La 2ème API est : http://ticher.stellar.org qui renvoi une liste des taux du XLM, dont on récupère le taux XLM/EUR. Puis lorsque les 2 taux sont récupérés, on calcule le taux XLM/CHF, puis on l'affiche sur le site et on s'en sert pour afficher les valeurs dans les 2 monnaies.

Pour avoir le bon taux de Lumens, on met à jour toutes les 5 minutes sont taux, car il est variable.

5.2.3 Code

```
//Le taux de change (XLM/CHF) est mis à jour toutes les 5 minutes
var k = schedule.scheduleJob('*/5 * * * *', function(){
    majTaux();
});
```

```
function majTaux(){
    var tauxLumensEuro;
    var tauxEuroCHF;
    var xmlhttp;
    // compatible with IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
            var data = JSON.parse(xmlhttp.responseText);
            tauxEuroCHF = data.rates.CHF;
            console.log("tauxEuroCHF  "+tauxEuroCHF);
            var xmlhttp2;
            // compatible with IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp2 = new XMLHttpRequest();
            xmlhttp2.onreadystatechange = function(){
                if (xmlhttp2.readyState == 4 && xmlhttp2.status == 200){
                    var data = JSON.parse(xmlhttp2.responseText);
                    data.forEach(function(obj) {
                        if(obj.Name == "XLM_EUR"){
                            tauxLumensEuro = obj.Price;
                        }
                    });
                    tauxLumensCHF = tauxLumensEuro * tauxEuroCHF;;

                }
            }
            xmlhttp2.open("GET", "http://ticker.stellar.org", true);
            xmlhttp2.send();
        }
    }
    xmlhttp.open("GET", "http://api.fixer.io/latest", true);
    xmlhttp.send();
```

5.3 Enregistrer les investissements

5.3.1 Problème

Le compte de gestion du site contient la somme totale de toutes les propositions en cours, donc le compte ne sait pas combien de Lumens appartiennent à quelles personnes.

5.3.2 Solution

Enregistrer chaque transaction qui est faite, pour avoir une liste des investissements et enregistrer également les clés publiques des utilisateurs ayant fait les propositions et celles des proposants.

Associer la liste des investissements à chaque proposition et les envoyer dans le JSON qui est enregistré dans une entry sur Cowaboo.

5.3.3 Code

```
function investir(){
    attendre(true);
    var secretKey = prompt("Veuillez entrez votre clé secrète s'il vous plait.", "");
    if (secretKey != null) {
        var periode = tabPeriodes.pop();
        var proposition = getProposition(id, periode.listePropositions);
        var utilisateur = getUtilisateur(sessPublicKey);
        var valeur = $('#nbLumens').val();
        var balance;
        var jqxhr =  $.get( "/balance");
        jqxhr.done(function( data ) {
            $.each(JSON.parse(data), function(i, obj) {
                balance = obj.balance;
            });
            if((balance-valeur)>20){
                var destinationKey = investProjectPublicKey;
                var dataString = 'secretKey='+ secretKey.toString() + '&destinationKey=' + destinationKey + '&valeur=' + (valeur-0.00001).toString();
                $.ajax({
                    type: "POST",
                    url: "/transaction",
                    data: dataString,
                    success: function(data) {
                        if(data=="error"){
                            alert("L'investissement n'a pas pu être pris en compte dû à un soucis d'un serveur.");
                            tabPeriodes.push(periode);
```

```
                        } else{
                            attendre(false);
                            proposition.listeInvestissements.push(new Investissement(sessPublicKey, valeur-0.01, moment().format('DD/MM/YYYY à HH:mm:ss'),
                                                                    data._links.transaction.href, sessEmail));
                            proposition.balance += valeur-0.01;
                            periode = setProposition(periode, proposition);
                            tabPeriodes.push(periode);
                            putPeriodes("investissement");
                            //alert("investissement réussi !");
                        }

                    },
                    error: function(data) {
                        attendre(false);
                        alert("L'investissement n'a pas pu être fait, dû à un souci d'un serveur");
                    }
                });
            }else{
                attendre(false);
                tabPeriodes.push(periode);
                alert("Votre compte doit garder 20 lumens minimum.");
            }

        });
        jqxhr.fail(function(data){
            attendre(false);
            balance = -1;
        });
    } else{
        attendre(false);
        alert("Veuillez rentrez une clé secrète valide s'il vous plait.");
    }
```

5.4 Distribuer les investissements automatiquement

5.4.1 Problème

On souhaite avoir un site qui fonctionne seul une fois lancé et avec un minimum d'intervention d'un administrateur. Donc à la fin de chaque période il faut que tous les Lumens soient distribués aux bonnes personnes, sans qu'une personne ne fasse l'opération.

5.4.2 Solution

Mettre en place un algorithme qui s'exécute le 1er de chaque mois et qui analyse chaque proposition de la période en cours, pour envoyer selon le total d'investissement de la proposition, les Lumens soit aux investisseurs, soit aux proposant. L'algorithme récupère les informations dont il a besoin pour exécuter les transactions et ensuite il lance chaque transaction en envoyant les Lumens depuis le compte de Gestion du site, vers le compte des utilisateurs correspondant.

5.4.3 Code

```
//fonction qui va être lancée tous les 1er du mois
var j = schedule.scheduleJob('0 0 1 * *', function(){
    initPeriodes("lancerTransactions");
});
```

```
//Récupère le JSON des périodes avec les propositions et leurs investissements et les converti en objet
function initPeriodes (nomFonction=""){
    attendre(true);
    tabPeriodes = [];
    var xmlhttp;
    xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
            var data = JSON.parse(xmlhttp.responseText);
            valueJSON = data.dictionary.entries;
            obj = valueJSON;
            for(var key in obj){
                hash = key;
                objJ = JSON.parse(obj[key].value);
                objJ.forEach(function(obj2){
                    tabPropositions = [];
                    obj2.listePropositions.forEach(function(obj3) {
                        tabInvestissements = [];
                        obj3.listeInvestissements.forEach(function(obj4){
                            tabInvestissements.push(new Investissement(obj4.accountId, obj4.montant, obj4.dateInvestissement, obj4.lienTransaction));
                        });
                        idMaxProposition = obj3.propositionId;
                        tabPropositions.push(new Proposition(obj3.nomProjet, obj3.dateDebutProjet, obj3.dateFinProjet, obj3.descriptionProjet,
                                                             obj3.propositionId, obj3.publicKey, obj3.balance, obj3.statut, obj3.listeInvestissements));
                    });
                    tabPeriodes.push(new Periode(obj2.numeroPeriode, obj2.dateDebutPeriode, obj2.dateFinPeriode, tabPropositions));
                })
            }
```

```
            //On sait quoi faire lorsque la requête est terminée
            switch(nomFonction) {
                case "afficherProposition":
                    afficherProposition();
                    break;
                case "afficherListePropositions":
                    afficherListePropositions();
                    break;
                case "lancerTransactions":
                    lancerTransactions();
                    break;
                case "afficherListeInvestissements":
                    afficherListeInvestissements();
            }
        } else if (xmlhttp.readyState == 4) {
            alert("il y a eu une erreur lors du chargement des propositions, veuillez s'il vous plait relancer la page.");
        }
    }
    xmlhttp.open("GET", "http://groups.cowaboo.net/group-coInvest/public/api/observatory?observatoryId=Propositions", true);
    xmlhttp.send();
}
```

```
//Fonction qui lance les transactions correspondantes aux résultats de chaque
function lancerTransactions(){
    var periode = tabPeriodes.pop();
    periode.listePropositions.forEach(function(obj){
        //si la balance de la proposition atteint l'objectif d'investissement, alors le compte de l'utilisateur
        //qui a fait la proposition est crédité des lumens
        if((obj.balance*tauxLumensCHF) >= (objectifInvestissement)){
            doTransaction(secretInvestProject, obj.publicKey, obj.balance.toString());
            obj.statut = "investissements obtenus";
            envoyerEmail(obj.email, '', obj.nomProjet, obj.balance);
            obj.listeInvestissements.forEach(function(obj2){
                envoyerEmail(obj2.email, false, obj.nomProjet, obj2.montant);
            });
        //si la balance de la proposition n'atteint pas l'objectif d'investissement, alors les investissements sont
        //renvoyé aux investisseurs
        }else{
            obj.listeInvestissements.forEach(function(obj2){
                doTransaction(secretInvestProject, obj2.publicKey, obj2.montant.toString());
                envoyerEmail(obj2.email, true, obj.nomProjet, obj2.montant);
            });
            obj.statut = "investissements rendus";
        }
    });
    tabPeriodes.push(periode);
```

```
    var d = new Date();
    //le mois renvoyé est un chiffre de 0 à 11, donc si on est en janvier, on aura 0, donc on fait +1 pour avoir 01/
    //et on fait +2 pour avoir le mois suivant pour la date fin, ce qui donne 02 (février).
    var mois = d.getMonth()+2;
    var jour = d.getDate();
    var heure = d.getHours();
    var minute = d.getMinutes();
    if (mois < 10) { mois = '0' + mois; }
    if (jour < 10) { jour = '0' + jour; }

    var dateDebut = jour+"/"+(d.getMonth()+1)+"/"+d.getFullYear();
    var dateFin = "01/"+mois+"/"+d.getFullYear();
    tabPeriodes.push(new Periode(periode.numeroPeriode+1, d.getFullYear, dateFin));
    majPeriodes(encodeURIComponent(JSON.stringify(tabPeriodes)), hash);
}
```

5.5 Prévenir les utilisateurs du résultat d'investissement

5.5.1 Problème

Sans venir régulièrement sur le site, les utilisateurs ne peuvent pas savoir si leurs Lumens investis leurs sont retournés ou ont été envoyé sur le compte du proposant. Ils doivent venir sur le site, regarder les propositions de projet auxquels ils ont envoyé des Lumens et regarder leur statut.

5.5.2 Solution

Récupérer l'email à chaque investissement et création de proposition, puis envoyer un email à chaque utilisateur expliquant si l'investissement leur a été rendu ou s'il a été envoyé sur le compte du proposant. Les proposants reçoivent également un email qui les prévient du résultat.

5.5.3 Code

```
function envoyerEmail(emailClient, retourInvestissement, nomProj, solde){
    var message ="";
    if(retourInvestissement==true){
        message = "<b> Le projet : "+nomProj+" n'a pas atteint l'objectif, donc votre investissement de "+solde+" XLM / "+
        (solde*tauxLumensCHF).toFixed(2)+" CHF retourne sur votre compte.<br><br>Si dans les 5-10 minutes qui suivent,"+
        "vous n'avez pas reçu la somme, veuillez contacter l'administrateur du site internet. Vous trouverez ses informations"+
        " de contact dans le pied de page du site internet. <br><br><br> InvestProject</b>"
    }else if(retourInvestissement==false){
        message = "<b> Le projet : "+nomProj+" a atteint l'objectif, donc votre investissement de "+solde+" XLM / "+
        (solde*tauxLumensCHF).toFixed(2)+" CHF a été envoyé sur le compte du proposant du projet. <br><br><br> InvestProject</b>"
    }else if(retourInvestissement=='trueProposant'){
        message = "<b> Votre projet : "+nomProj+" a atteint l'objectif, donc l'investissement de "+solde+" XLM / "+
        (solde*tauxLumensCHF).toFixed(2)+" CHF a été envoyé sur votre compte. <br><br><br> InvestProject</b>"
    }else {
        message = "<b> Votre projet : "+nomProj+" n'a pas atteint l'objectif, donc l'investissement de "+solde+" XLM / "+
        (solde*tauxLumensCHF).toFixed(2)+" CHF a été retourné sur les comptes des investisseurs. <br><br><br> InvestProject</b>"
    }

    var mailOptions = {
        from: email,
        to: emailClient,
        subject: "Résultat d'un investissement sur InvestProject",
       // text: req.body.message,
        html: message
    };

    transporter.sendMail(mailOptions, function(error, info){
        if(error){
            return console.log(error);
        }
        console.log('Message sent: ' + info.response);
    });

}
```

6. Propositions de futures améliorations

Toute application peut ou va être amélioré. Durant le développement d'InvestProject, plusieurs idées me venaient à l'esprit, sans avoir le temps de les implémenter, donc voici une liste d'améliorations possible pour le futur.

6.1 Vérification du contenu

Le site est fonctionnel en considérant que tous les utilisateurs sont de bonne foi, mais un utilisateur malveillant pourrait créer plusieurs propositions de projet pour spammer le site. Donc, il faudrait implémenter une règle, comme quoi chaque adresse à le droit de créer uniquement une proposition.

6.2 Contrôle des transactions

Les transactions faites par les utilisateurs et les transactions faites par le site internet fonctionnent correctement. L'investissement des utilisateurs est enregistré uniquement lorsque la transaction est réussie. Mais, la fonction automatique qui lance plusieurs transactions en même temps ne repère pas lorsqu'une transaction a eu un problème. Donc, si un des serveurs utilisés pour effectuer les transactions ne fonctionne pas, ça ne sera pas pris en compte et peut-être certains utilisateurs n'auront pas récupéré leurs Lumens.

Il serait donc intéressant de mettre en place un système, qui met à jour un statut de chaque investissement, pour savoir lorsqu'il y a eu un problème dans l'une des transactions et dans ce cas, elles seraient relancées automatiquement au maximum 3 fois et si au bout de 3 fois, il y a encore eu un problème, un email serait envoyé à l'administrateur du site internet.

Mettre en place une partie amélioration du site ou les personnes peuvent proposer des améliorations du site ou les utilisateurs peuvent voter.

6.3 Vote pour améliorer / modifier le site internet

Une partie amélioration du site internet serait intéressante à mettre en place, pour que les utilisateurs participent à l'amélioration du site internet. Il y aurait un endroit ou pendant chaque période, il y aurait quelques propositions d'amélioration et les utilisateurs auront une possibilité de voter. Ce qui augmenterait leur confiance envers le site internet et les impliquerait plus.

6.4 Choix de l'objectif

L'objectif actuel est de 3'000 CHF pour toutes les propositions de projet. Il faudrait permettre aux utilisateurs, entre deux périodes, de choisir un objectif et ensuite le

système effectuerait une moyenne des propositions pour définir le prix de l'investissement à atteindre pour la période suivante.

Bien sûr, pour éviter qu'il y ait des propositions complètements dérisoire, une fourchette de prix serait proposée en fonction de l'objectif de la période précédente. Par exemple : le maximum possible, serait le double du prix précédant et le minimum serait sa moitié.

Plusieurs salons d'investissement

Tous les projets n'ont pas la même ampleur, donc faire plusieurs salons qui ont un niveau d'objectif de taille différente, permettrai d'augmenter les possibilités de propositions de projets. Il faudrait évidemment ne pas faire un trop grand nombre de salon, sinon ça n'aurait plus d'utilité, mais un nombre de 3 salons serait un bon chiffre. Par exemple, un salon pour les petits projets qui demanderait 3'000 CHF d'investissements, un moyen qui demanderait 15'000 CHF et un grand qui demanderait 50'000 CHF.

6.5 Plusieurs crypto-monnaies

Les investissements se font actuellement en Lumens, un ajout de plusieurs crypto-monnaies laisserait plus de liberté aux utilisateurs. Lors de l'investissement, le choix de la crypto-monnaie serait proposé, par contre le site les convertiraient toutes en une seul, lorsqu'un projet récupère son investissement.

6.6 Gestion du compte d'utilisateur

La seule information que voient les utilisateurs sur le site, est la balance de leur compte. Donc, il faudrait rajouter une page qui leur permet de voir l'historique de leurs investissements et l'historique des propositions qu'ils ont fait. Actuellement ils peuvent accéder à ces informations en sélectionnant les périodes des propositions et en allant sur les moniteurs, mais ils doivent se souvenir des périodes et des projets auxquels ils ont participés.

Il pourrait également y avoir un système de notification lors de la fin de la période pour prévenir l'utilisateur de ce qui a été fait avec son investissement, si il a été envoyé au proposant ou si il lui a été retourné. Actuellement, les utilisateurs sont prévenus par email.

6.7 Affichage sur Cowaboo

Il serait intéressant pour toucher plus de monde, sachant que les comptes d'InvestProject, sont des comptes Cowaboo, de publier les propositions de projet sur Cowaboo. C'est-à-dire, que lorsqu'une personne créer une proposition, une entry serait créée sur Cowaboo avec sa description, ainsi qu'un lien amenant directement sur la proposition sur InvestProject.

7. Conclusion

Ce travail de Bachelor m'a fait très peur au début, car je me lançais dans beaucoup de choses nouvelles pour moi en termes de développement. Et plus les choses avançait et plus la peur laissait place à place à de la nouvelle connaissance.

J'ai appris beaucoup de choses durant ce travail de Bachelor et je suis convaincu que ces choses vont beaucoup me servir pour l'avenir. Mais, il y a également beaucoup de moments où je me servais de ce que j'ai appris dans cette école, dans les différents cours avec les professeurs qui les enseignent.

Durant cette année scolaire 2016/2017 j'ai commencé à entendre parler de la Blockchain à une conférence qui s'est déroulé à Genève, où un de nos professeurs a voulu nous y emmener, car il trouvait ce sujet très intéressant et très utile pour nous, car nous allions l'aborder au semestre suivant.

De plus, j'ai beaucoup vu de sujets en rapport avec la Blockchain, car j'ai eu des présentations à faire durant le semestre de printemps en 2017, et donc j'ai participé à quelques conférences pour m'instruire sur le sujet et avoir une vision plus claire.

Donc, quand j'ai reçu cette proposition de travail de Bachelor, j'étais très hésitant au début, car je devais entrer dans un domaine qui est nouveau pour moi et j'aime bien avoir la vision des choses que je vais faire, et là je n'arrivais pas à visionner comment j'allais m'y prendre.

Je suis content de ce que j'ai fait, même si je reste déçu que certaines choses n'aient pas fonctionné correctement ou j'ai dû appliquer des plans B.

J'ai fait plusieurs projets depuis que je suis dans le domaine de l'informatique, et je peux dire que celui-ci a été le plus différents de tous et le plus compliqué, mais aussi le plus instructif.

Ça aurait été plus facile de faire un développement dans un cadre plus familier pour moi, mais ce n'est pas en restant dans son confort qu'on avance, il faut apprendre et faire de nouvelles choses, donc je suis content d'avoir accepté ce travail de Bachelor, malgré les difficultés que j'ai rencontré durant le développement.

Bibliographie

HOTTOT, Kevin, 2016. The DAO : un pirate dérobe 50 millions de dollars, la contre-attaque se prépare. Nextinpact.com[en ligne]. 22 juin 2016. [Consulté le 12 juillet 2017]. Disponible à l'adresse : https://www.nextinpact.com/news/100336-the-dao-pirate-derobe-50-millions-dollars-contre-attaque-se-prepare.htm

POLROT, Simon, 2017. The DAO : post mortem. Ethereum-France.com[en ligne]. 24 janvier 2017. 7 juillet 2017 [Consulté le 12 juillet 2017]. Disponible à l'adresse : https://www.ethereum-france.com/the-dao-post-mortem/

VESSENES, Peter, 2016. Deconstructing theDAO attack : A Brief Code Tour. Vessenes.com[en ligne]. 18 juin 2016. [Consulté le14 juillet 2017]. Disponible à l'adresse : http://vessenes.com/deconstructing-thedao-attack-a-brief-code-tour/

DAIAN, Phil, 2016. Analysis of the DAO exploit. Hackingdistributed.com[en ligne]. 18 juin 2016. [Consulté le 17 juillet 2017]. Disponible à l'adresse : http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

POLROT, Simon, 2016. To fork or not to fork, telle est la question !. ethereum-France.com[en ligne]. 27 juin 2016. 30 mai 2017. [Consulté le 17 juillet 2017]. Disponible à l'adresse : https://www.ethereum-france.com/to-fork-or-not-to-fork-telle-est-la-question/

Jean-Luc, 2016. Ethereum vs Ethereum Classic. Bitcoin.fr[en ligne]. 27 juillet 2016. 28 juillet 2016. [Consulté le 17 juillet 2017]. Disponible à l'adresse : https://bitcoin.fr/ethereum-vs-ethereum-classic/

wikipedia, 2014. Stellar (payment network). Wikipedia.org[en ligne]. 4 septembre 2014. 17 septembre 2017. [Consulté le 16 septembre 2017]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Stellar_(payment_network)&action=history

ERDOGAN, Onur & ATTIAS, Léa. CoWaBoo, un projet qui se concrétise. Ageneve.net[en ligne]. 20 mai 2015. [Consulté le 16 septembre 2017]. Disponible à l'adresse : http://www.ageneve.net/leayouyou/2015/05/20/cowaboo/

Annexe 1 : Règlement du site

Ceci est le règlement et fonctionnement d'InvestProject. Pour pouvoir entrer sur le site, vous allez devoir le lire attentivement et l'accepter pour être conscient des tenants et aboutissements de vos actes sur le site.

Toutes les règles qui sont implémentées par du code, sont associées à une image qui montre la partie du code qui les concerne. Pour regarder tout le code d'InvestProject, aller sur son Github à ce lien : https://github.com/saidiamir/InvestProject2

Règle N°1

Dans le site, tous les investissements et les récupérations d'investissement se font avec la crypto-monnaie « Lumens » (XLM).

Image

Règle N°2

Pour recharger votre compte, il faudra faire la démarche qui est décrite dans le pdf à votre disposition sur le site, à la page « Ajouter des Lumens ».

Image

Règle N°3

Votre compte ne doit pas descendre en dessous de 20 Lumens pour qu'il reste actif, donc toutes transactions ramenant votre solde en dessous de ce seuil seront bloquées, pour permettre le bon fonctionnement de votre compte.

Image

```
var balance;
var jqxhr =  $.get( "/balance");
jqxhr.done(function( data ) {
    $.each(JSON.parse(data), function(i, obj) {
        balance = obj.balance;
    });
    if((balance-valeur)>20){
```

Règle N°4

Chaque mois, il y a une période de proposition de projet allant du début du mois, jusqu'au début du mois suivant. Lorsque la période est terminée, les Lumens sont redistribués et une nouvelle période est lancée.

Image

```
//fonction qui va être lancée tous les 1er du mois
var j = schedule.scheduleJob('0 0 1 * *', function(){
    initPeriodes("lancerTransactions");
});
//Fonction qui lance les transactions correspondantes aux résultats de chaque
function lancerTransactions(){
    var periode = tabPeriodes.pop();
    periode.listePropositions.forEach(function(obj){
        //si la balance de la proposition atteint l'objectif d'investissement, alors le compte de l'utilisateur
        //qui a fait la proposition est crédité des lumens
        if((obj.balance*tauxLumensCHF) >= (objectifInvestissement)){
            doTransaction(secretInvestProject, obj.publicKey, obj.balance.toString());
            obj.statut = "investissements obtenus";
            envoyerEmail(obj.email, 'trueProposant', obj.nomProjet, obj.balance);
            obj.listeInvestissements.forEach(function(obj2){
                envoyerEmail(obj2.email, false, obj.nomProjet, obj2.montant);
            });
        //si la balance de la proposition n'atteint pas l'objectif d'investissement, alors les investissements sont
        //renvoyé aux investisseurs
        }else{
            envoyerEmail(obj.email, 'falseProposant', obj.nomProjet, obj.balance);
            obj.listeInvestissements.forEach(function(obj2){
                doTransaction(secretInvestProject, obj2.publicKey, obj2.montant.toString());
                envoyerEmail(obj2.email, true, obj.nomProjet, obj2.montant);
            });
            obj.statut = "investissements rendus";
        }
    });
    tabPeriodes.push(periode);
    var d = new Date();
    //le mois renvoyé est un chiffre de 0 à 11, donc si on est en janvier, on aura 0, donc on fait +1 pour avoir 01/
    //et on fait +2 pour avoir le mois suivant pour la date fin, ce qui donne 02 (février).
    var mois = d.getMonth()+2;
    var jour = d.getDate();
    var heure = d.getHours();
    var minute = d.getMinutes();
    if (mois < 10) { mois = '0' + mois; }
    if (jour < 10) { jour = '0' + jour; }

    var dateDebut = jour+"/"+(d.getMonth()+1)+"/"+d.getFullYear();
    var dateFin = "01/"+mois+"/"+d.getFullYear();
    tabPeriodes.push(new Periode(periode.numeroPeriode+1, d.getFullYear, dateFin));
    majPeriodes(encodeURIComponent(JSON.stringify(tabPeriodes)), hash);
}
```

Règle N°5

L'objectif commun à toutes les propositions de projet est de 3'000CHF, pour que le proposant récupère l'investissement il faut que le total investi atteigne cet objectif.

Image

```
//objectif en chf.
var objectifInvestissement = 3000;
//si la balance de la proposition atteint l'objectif d'investissement, alors le compte de l'utilisateur
//qui a fait la proposition est crédité des lumens
if((obj.balance*tauxLumensCHF) >= (objectifInvestissement)){
    doTransaction(secretInvestProject, obj.publicKey, obj.balance.toString());
    obj.statut = "investissements obtenus";
    envoyerEmail(obj.email, 'trueProposant', obj.nomProjet, obj.balance);
    obj.listeInvestissements.forEach(function(obj2){
        envoyerEmail(obj2.email, false, obj.nomProjet, obj2.montant);
    });
```

Règle N°6

Lorsque vous investissez sur une proposition de projet, Les lumens investis sont transférés sur le compte de gestion du site et y resteront jusqu'à la fin de la période en cours.

Image

```
var destinationKey = investProjectPublicKey;
var dataString = 'secretKey='+ secretKey.toString() + '&destinationKey=' + destinationKey + '&valeur=' + (valeur-0.00001).toString();
```

Règle N°7

Lorsqu'un projet auquel vous avez investi des Lumens, n'atteint pas l'objectif, les Lumens vous seront rendus, moins une taxe de 0,1% qui couvre les frais de transactions et de fonctionnement du site internet.

Image

```
proposition.listeInvestissements.push(new Investissement(sessPublicKey, valeur-(valeur*0.1%), moment().format('DD/MM/YYYY à HH:mm:ss'),
                                      data._links.transaction.href, sessEmail));
```

Règle N°8

Lorsqu'un projet atteint l'objectif, le proposant récupère le total d'investissement, auquel les taxes ont déjà été soustraites.

Image

```
var periode = tabPeriodes.pop();
periode.listePropositions.forEach(function(obj){
    //si la balance de la proposition atteint l'objectif d'investissement, alors le compte de l'utilisateur
    //qui a fait la proposition est crédité des lumens
    if((obj.balance*tauxLumensCHF) >= (objectifInvestissement)){
        doTransaction(secretInvestProject, obj.publicKey, obj.balance.toString());
        obj.statut = "investissements obtenus";
        envoyerEmail(obj.email, 'trueProposant', obj.nomProjet, obj.balance);
        obj.listeInvestissements.forEach(function(obj2){
            envoyerEmail(obj2.email, false, obj.nomProjet, obj2.montant);
        });
```

Règle N°9

Il est important de savoir que les Lumens n'ont pas une valeur fixe en CHF, c'est pourquoi sur le site internet il y a le taux XLM / CHF disponible qui est mis à jour toutes les 5 minutes et il y a également l'affichage des totaux d'investissements des propositions en XLM et CHF.

Image

```
//Le taux de change (XLM/CHF) est mis à jour toutes les 5 minutes
var k = schedule.scheduleJob('*/5 * * * *', function(){
    majTaux();
});
function majTaux(){
    var tauxLumensEuro;
    var tauxEuroCHF;
    var xmlhttp;
    // compatible with IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
            var data = JSON.parse(xmlhttp.responseText);
            tauxEuroCHF = data.rates.CHF;
            console.log("tauxEuroCHF  "+tauxEuroCHF);
            var xmlhttp2;
            // compatible with IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp2 = new XMLHttpRequest();
            xmlhttp2.onreadystatechange = function(){
                if (xmlhttp2.readyState == 4 && xmlhttp2.status == 200){
                    var data = JSON.parse(xmlhttp2.responseText);
                    data.forEach(function(obj) {
                        if(obj.Name == "XLM_EUR"){
                            tauxLumensEuro = obj.Price;
                        }
                    });
                    tauxLumensCHF = tauxLumensEuro * tauxEuroCHF;;

                }
            }
            xmlhttp2.open("GET", "http://ticker.stellar.org", true);
            xmlhttp2.send();
        }
    }
    xmlhttp.open("GET", "http://api.fixer.io/latest", true);
    xmlhttp.send();
}
```

En cliquant sur accepter, vous confirmez que vous avez bien lu et compris le règlement et fonctionnement du site internet.

Annexe 2 : document « AjouterDesLumens.doc »

12/10/2017

Alimenter son compte en Lumens

[Sous-titre du document]

Amir SAIDI (HES)
INVESTPROJECT

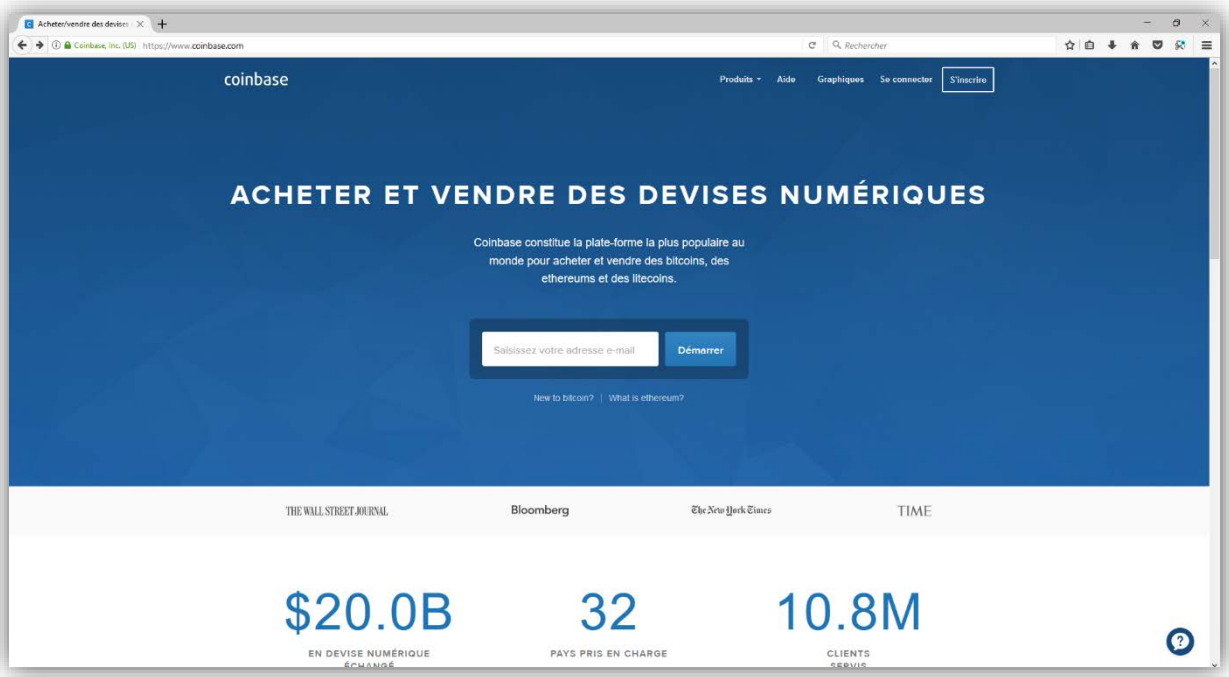
Table des matières

Introduction

Ce document est là pour vous aider dans la démarche d'alimentation de votre compte en Lumens. Il y a plusieurs possibilités pour faire cette démarche, je vous propose une façon de faire, mais si vous connaissez une autre méthode vous pouvez le faire, tant qu'au final votre compte sur le site contienne des Lumens.
Le tutoriel est fait sur les sites qui sont en anglais, donc si chez vous le site est en français, ne vous inquiétez pas, les boutons, onglets et champs, se trouvent à la même position.
L'alimentation de votre compte se fait en 2 étapes. La première étape consiste à acheter des Bitcoin pour ensuite les utiliser pour acquérir des Lumens. Si vous possédez déjà des Bitcoin et vous souhaitez les utiliser pour acheter des Lumens, vous pouvez passer directement à la partie 2. Remplacer juste les étapes où Coinbase est affiché, par votre site où vous avez vos Bitcoin.
Si votre site est un site d'échange et contient les Lumens (XML ou STR), vous pouvez directement les acheter dessus et les envoyer à votre compte sur InvestProject. Regardez tout de même la partie 2 pour être sûr de ne pas vous trompez.

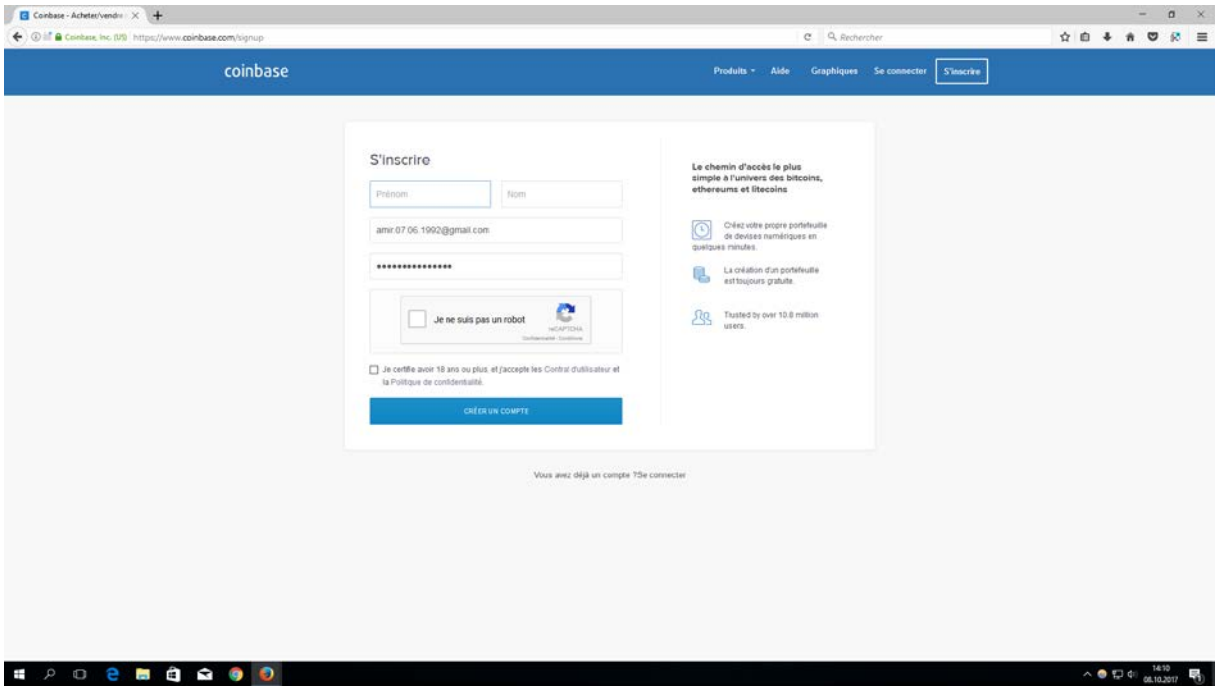
Partie 1 : Acheter des Bitcoin avec de l'argent réel.

Se rendre sur Coinbase



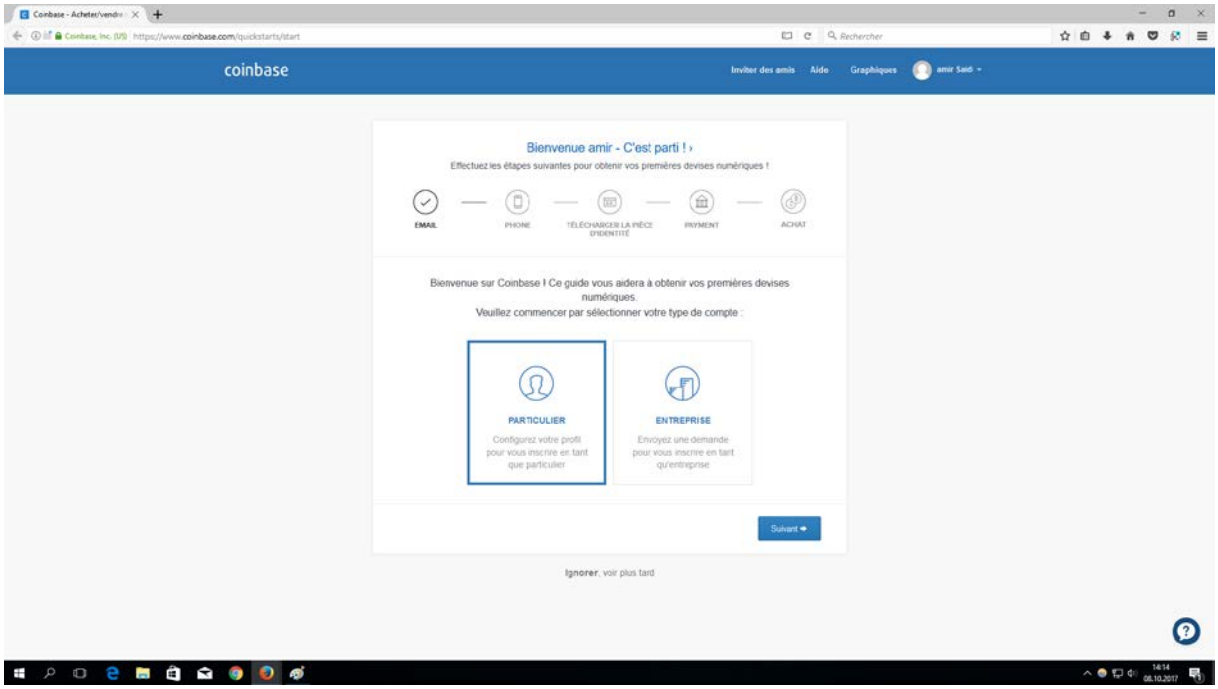
Pour commencer vous allez vous rendre sur https://www.coinbase.com. Coinbase est un site qui permet d'acheter des Bitcoin, des ether et des litecoin avec de l'argent réel, mais il permet également de faire l'inverse, c'est-à-dire vendre ces crypto-monnaies contre de l'argent réel.

S'inscrire à Coinbase



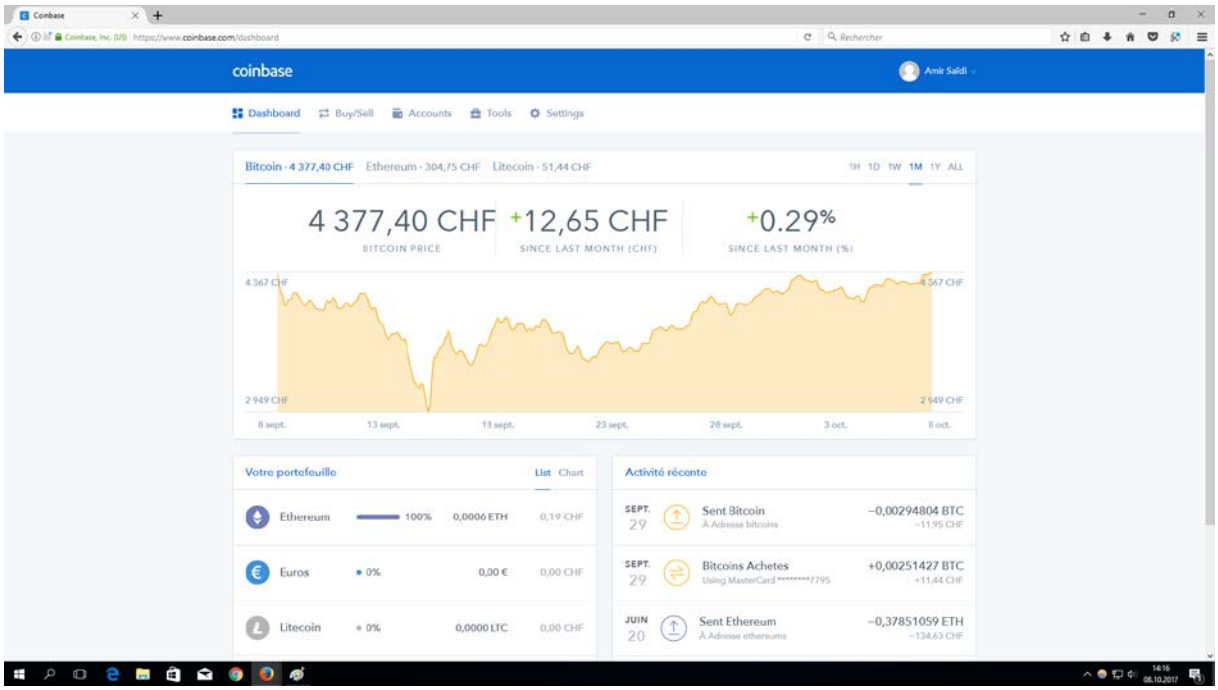
Vous allez cliquer sur « s'inscrire » en haut à gauche du site et vous allez remplir le formulaire avec vos informations. Lorsque vous l'aurez rempli, un email sera envoyé à l'adresse que vous avez mentionnée et vous devrez l'ouvrir et cliquer sur le bouton « vérifier l'adresse e-mail ».

Vérifier son compte



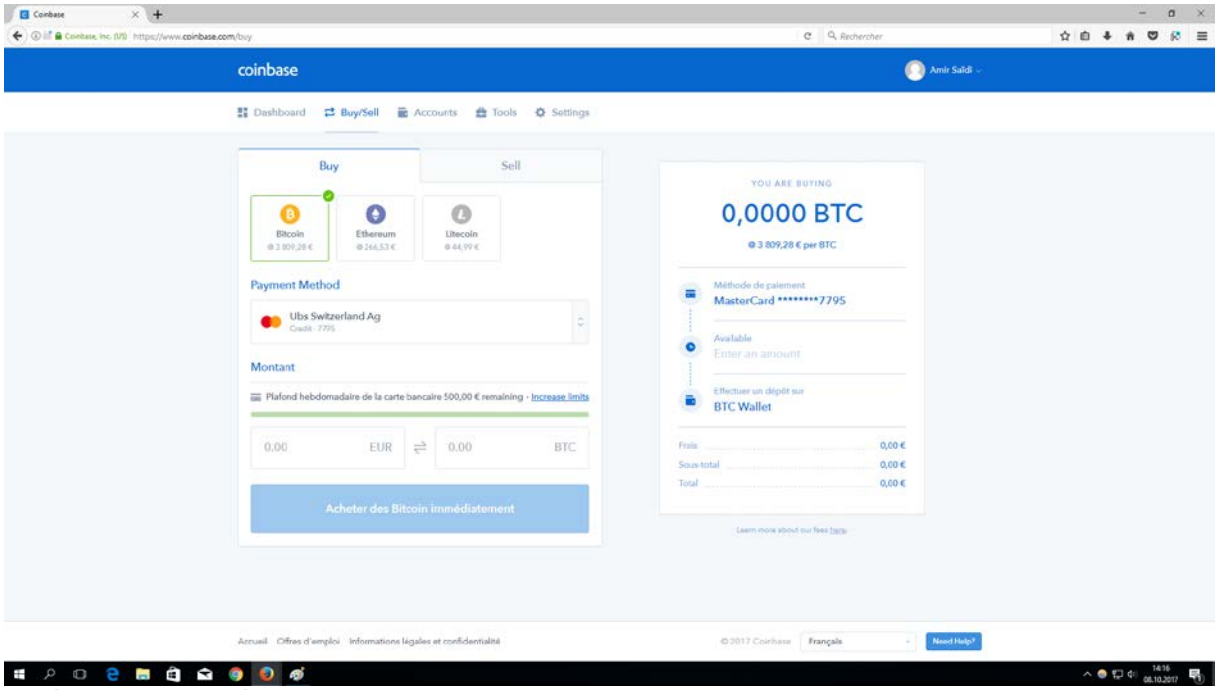
Lorsque vous aurez confirmé votre adresse e-mail, vous arriverez sur la page ci-dessus. Vous aurez plusieurs étapes de vérification à suivre, tout ceci est mis en place par Coinbase par soucis de sécurité. Certaines vérifications prennent du temps, comme la vérification de l'identité. Donc, vous pouvez en attendant, faire le début de la partie 2, c'est-à-dire l'inscription au site d'échange de monnaies.

Connexion à Coinbase



Lorsque vous aurez remplis toutes les étapes de vérifications, vous pourrez vous connecter sur le site et vous arriverez sur cette page d'accueil. Vous allez acheter des Bitcoin, donc vous allez cliquer sur « Buy/Sell ».

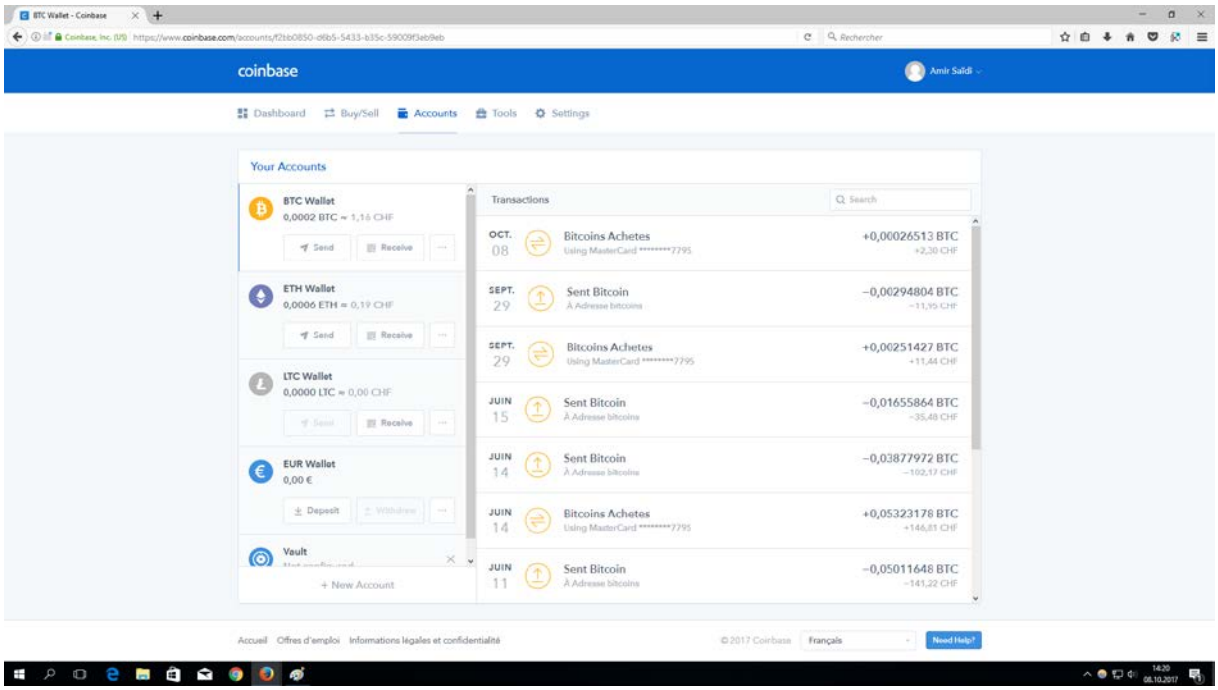
Achat de Bitcoin



Après avoir cliqué sur « Buy/Sell », vous arrivez sur la page ci-dessus. Vous allez sélectionner Bitcoin avec le logo jaune, et ensuite vous allez renseigner votre méthode de paiement et choisir combien d'argent souhaiter vous utiliser pour acheter des Bitcoin, ou si vous remplissez le champ Bitcoin, vous pouvoir choisir combien de Bitcoin vous souhaitez acheter.

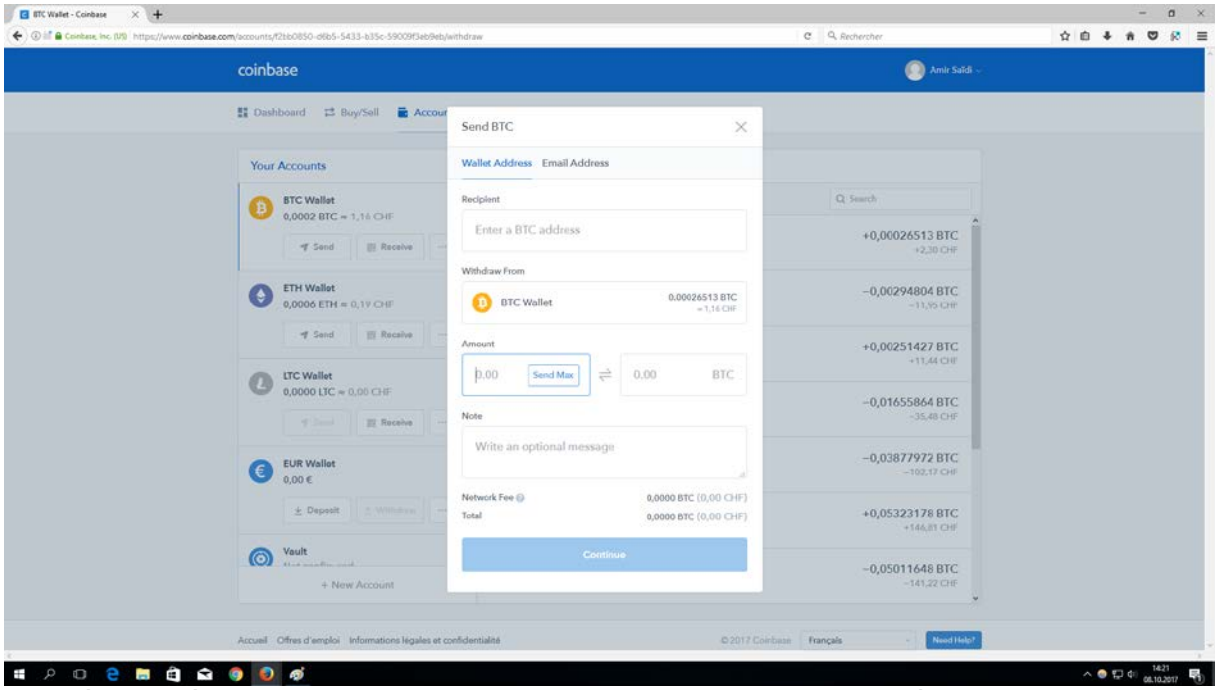
Lorsque vous aurez bien tout remplis, vous cliquerez sur le bouton en bas du formulaire : « Acheter des Bitcoin immédiatement ».

Voir son BTC Wallet



Lorsque le paiement ce sera déroulé correctement, vous pourrez voir en cliquant sur « Accounts » que votre BTC Wallet possède des BTC (Bitcoin). Vous devez cliquer sur le bouton « Send » juste en dessous de BTC Wallet.

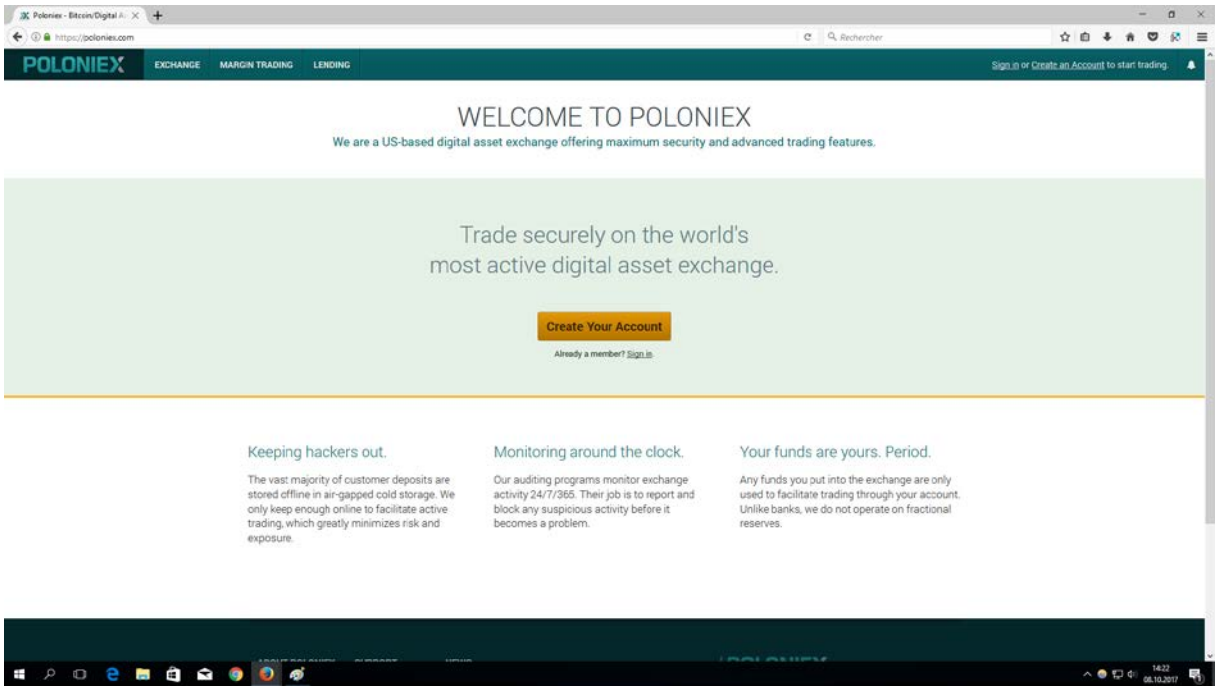
Formulaire d'envoi de Bitcoin



C'est à cette étape que vous allez envoyer des Bitcoin sur le site d'échange ou vous allez acheter des Lumens. Pour pouvoir remplir le champ « Recipient », vous aller devoir récupérer l'adresse depuis Poloniex, donc passez à la partie 2.

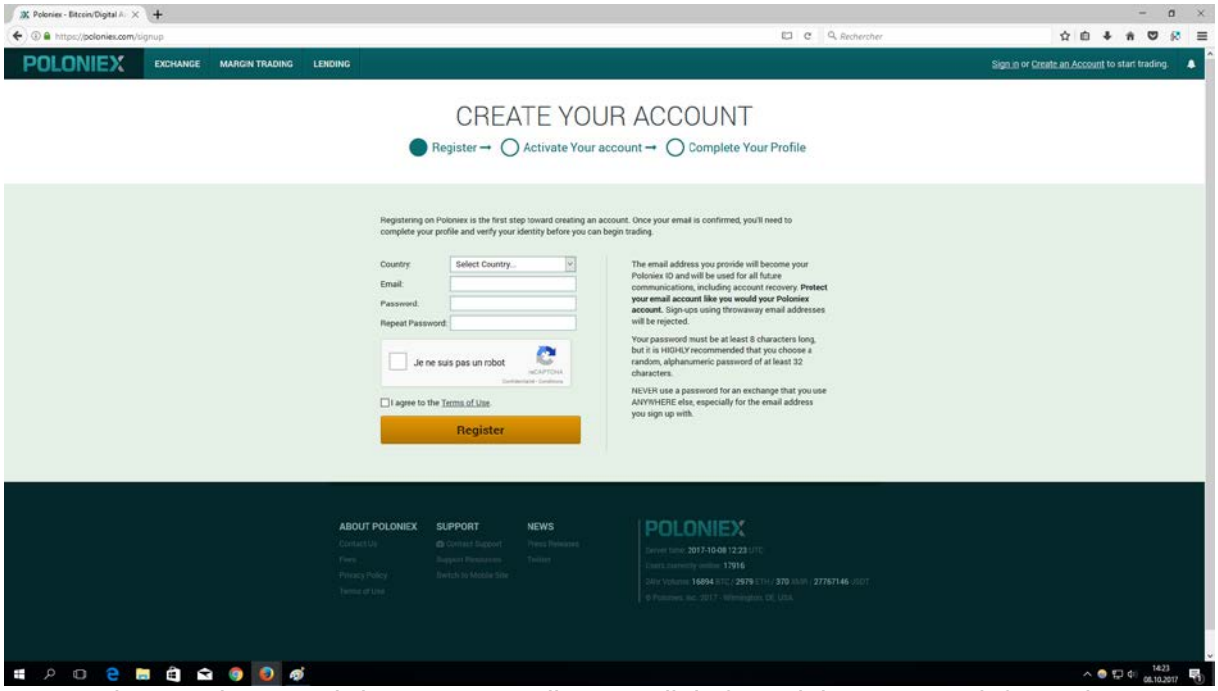
Partie 2 : Acheter des Lumens avec des Bitcoin.

Se rendre sur Poloniex



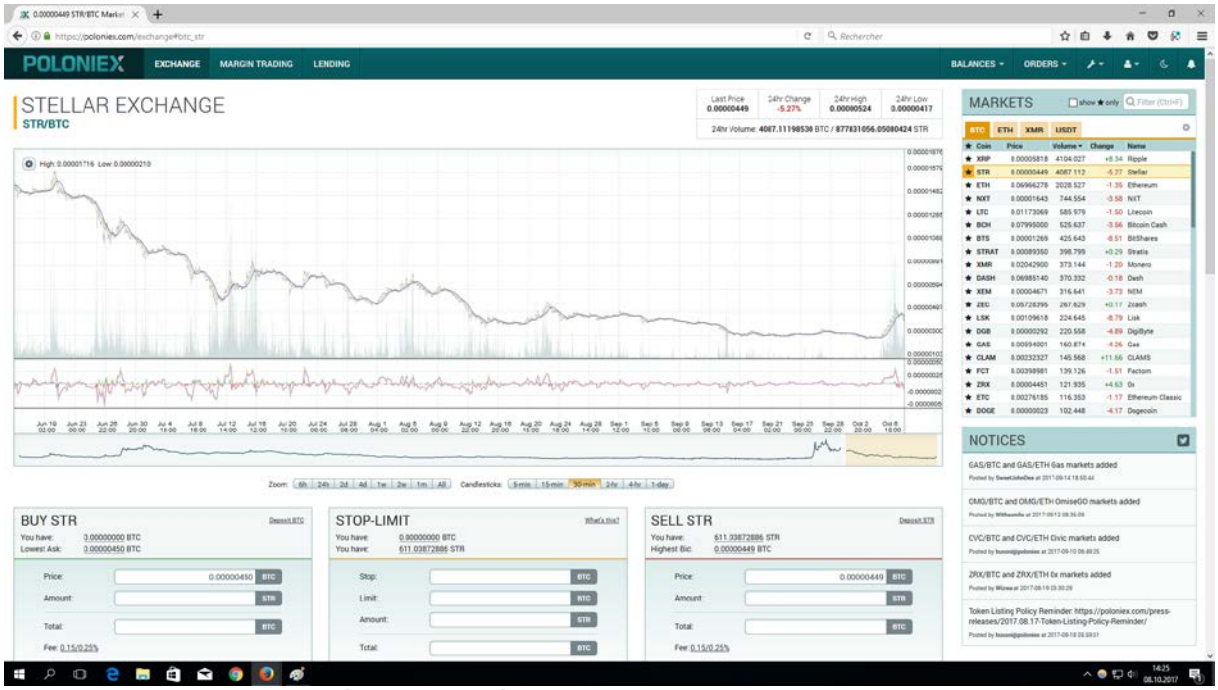
Pour commencer, vous allez vous rendre sur https://poloniex.com/. Poloniex est un site d'échange de crypto-monnaies. Il permet d'acheter différentes crypto-monnaies, avec 4 crypto-monnaies différentes. Le Bitcoin a plus de choix de crypto-monnaies, dont le STR (Stellar) qui correspond aux Lumens, sur d'autres sites d'échange, il peut avoir une autre abréviation qui est : XML. Le XML et le STR sont les Lumens, il n'y a aucune différences entre les 2, c'est juste une appellation différente.
Vous allez cliquer sur « Create Your Account ».

S’inscrire sur Poloniex



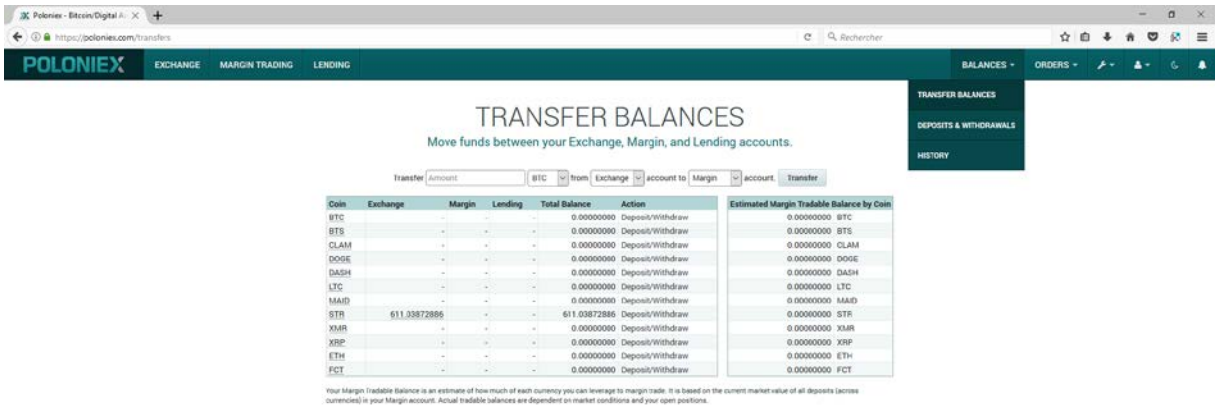
Vous arrivez sur la page ci-dessus. Vous allez remplir le formulaire avec vos informations et ensuite vous allez suivre les étapes.

Connexion à Poloniex



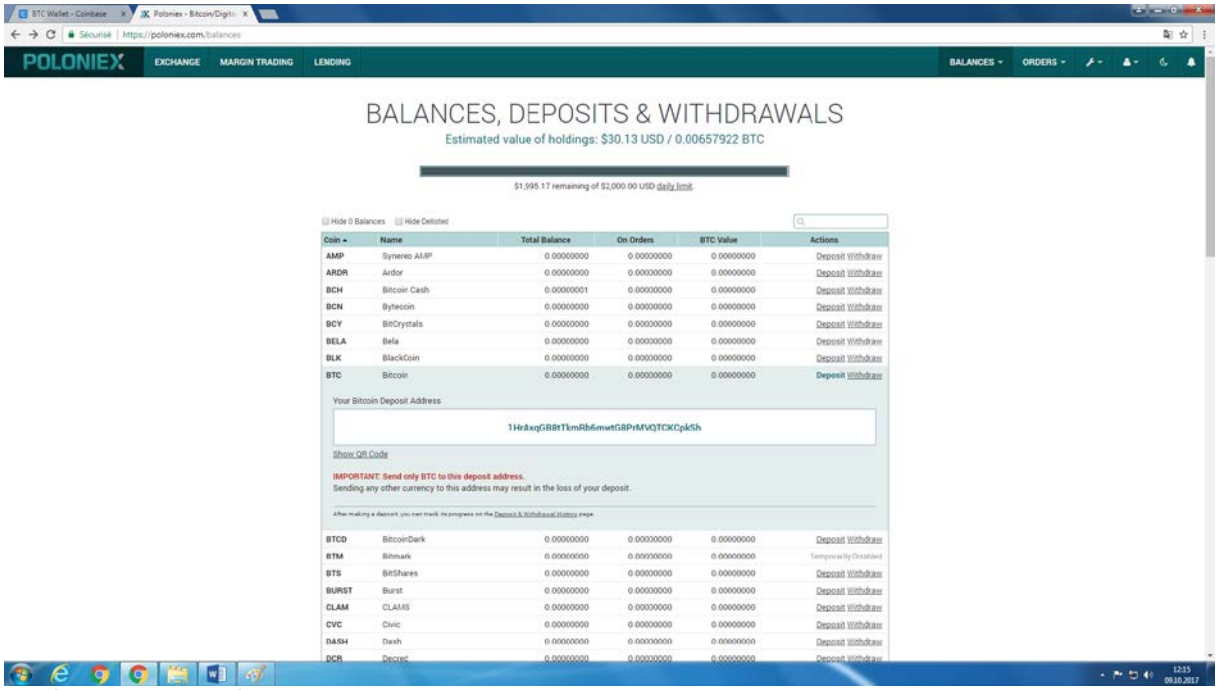
Lorsque vous avez terminé toutes les étapes d’inscription, vous allez vous connecter sur le site et allez arriver sur la page d’accueil comme ci-dessus. Glissez votre souris sur « BALANCES » et ensuite cliquez sur « TRANSFER BALANCES ». Si vous ne comprenez pas où se trouve « TRANSFER BALANCE », regarder l’image suivante.

Voir les Transfer Balances



Ici vous aurez une liste de crypto-monnaies, vous allez chercher la ligne où il y a écrit BTC dans la 1ère colonne. Lorsque vous avez repéré la ligne BTC, vous allez cliquer sur « Deposit/Withdraw » se situant à la même ligne que BTC.

Récupérer l'adresse des Bitcoin



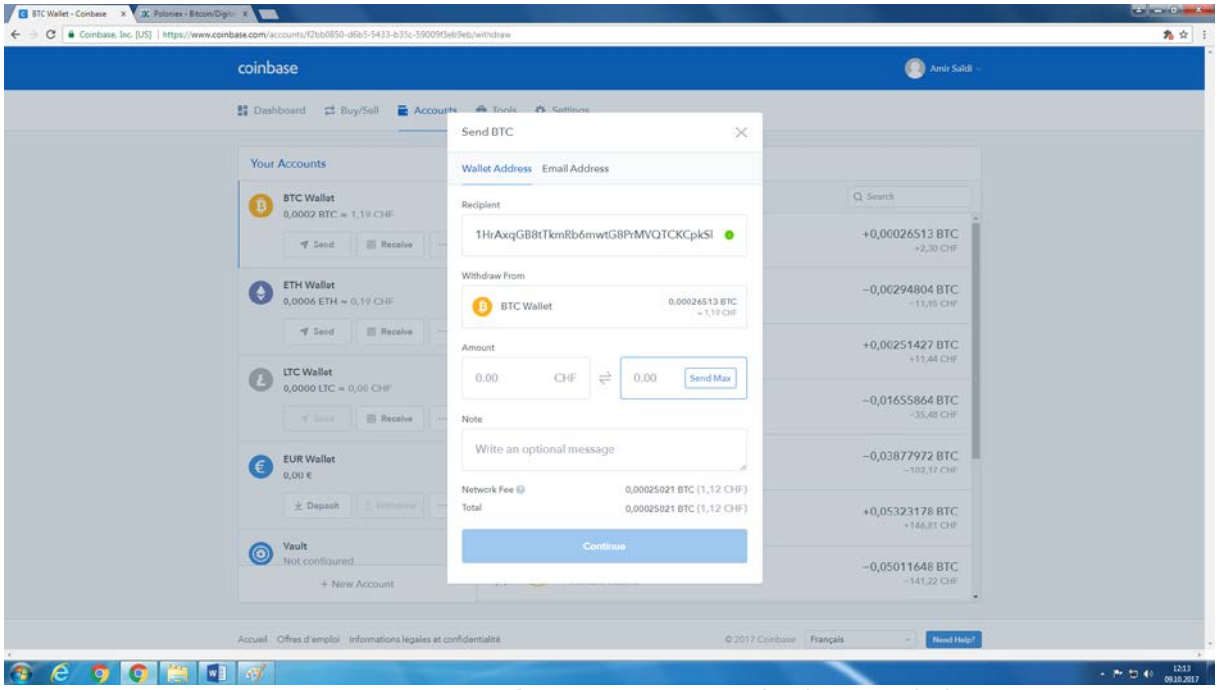
Après avoir cliqué sur « Deposit/Withdraw » vous allez arriver sur cette page avec la ligne BTC déployée.

Si elle n'affiche pas l'adresse, vérifier que dans la colonne action, c'est bien Deposit qui est sélectionné et non Withdraw. Si vous êtes bien sur Deposit, il y a une phrase où il devrait y avoir un lien avec écrit show adress. Cliquez sur ce lien et normalement vous devrez voir l'adresse apparaître comme dans l'image ci-dessus.

L'adresse qui est affichée est l'adresse qu'il faut copier pour la mettre dans le champ recipient du site Coinbase. Vous avez pu apercevoir à la fin de la partie 1, n'y retournez

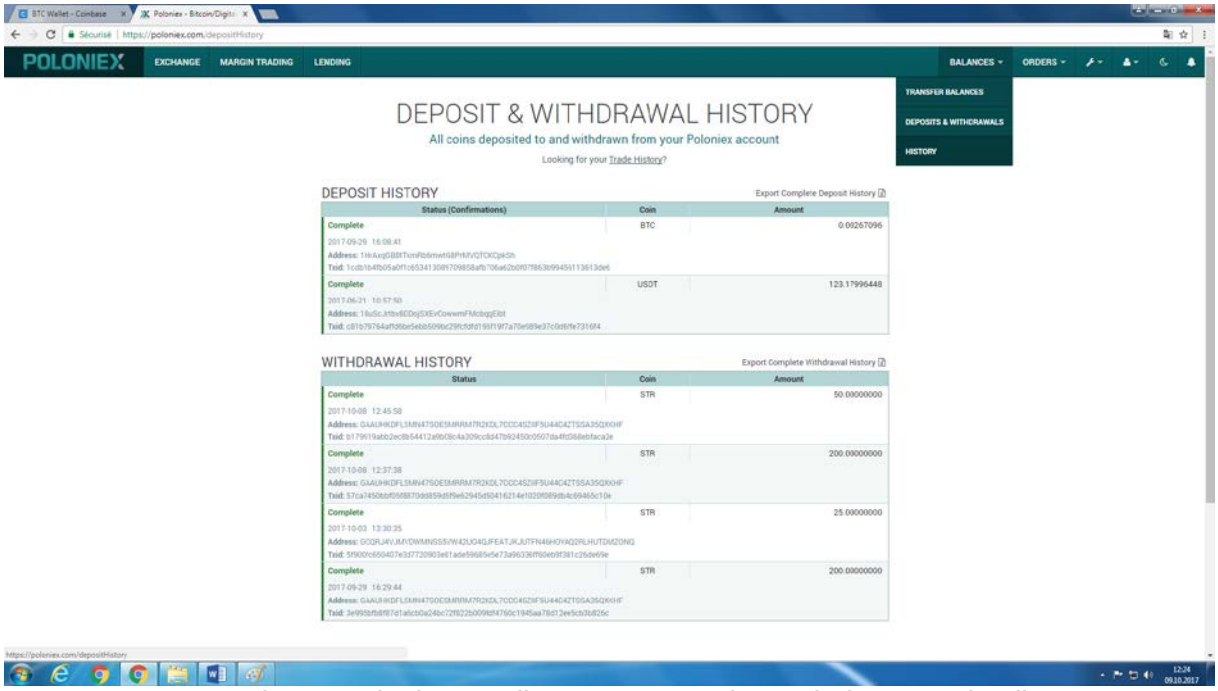
pas ! L'image suivante vous montre la même chose, avec le champ rempli avec l'adresse fournit sur Poloniex.

Remplir le formulaire d'envoi de Bitcoin



Vous allez sur Coinbase en ayant copié l'adresse fournie à l'étape précédente, et vous allez dans « Account » et sous BTC Wallet vous cliquer sur « Send ». Le formulaire comme sur l'image ci-dessus, va apparaître et vous devrez remplir le champ « Recipient » avec l'adresse que vous avez prise depuis Poloniex. Lorsque vous cliquez dans le champ de droite de « Amount », il y a un petit bouton « Send Max » qui y apparait, en cliquant dessus il va remplir le champ avec tous les Bitcoin que vous possédez. Si vous souhaitez envoyer tous vos Bitcoin vous cliquez dessus, sinon vous précisez le nombre de Bitcoin que vous voulez envoyer.
Lorsque vous avez rempli les champs « Recipient » et « Ammount », vous pouvez cliquer sur le bouton « Continue ».

Voir l'historique de l'envoi de Bitcoin



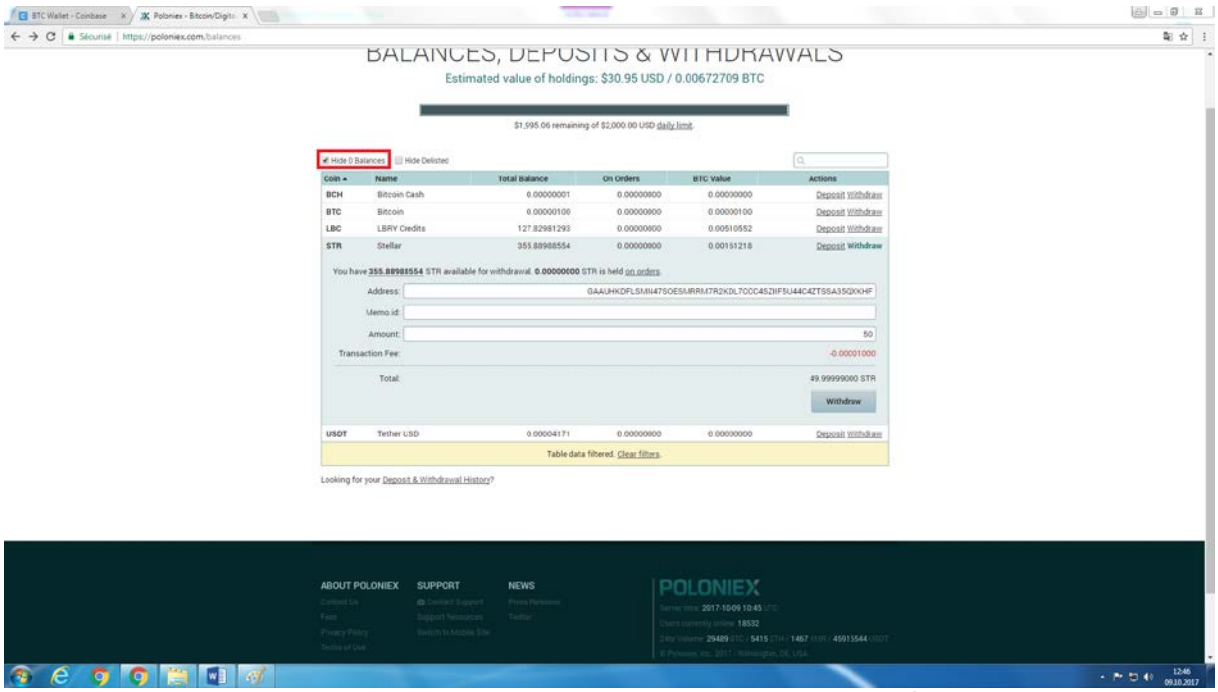
Vous pouvez revenir sur Poloniex et glisser votre souris sur balance, puis cliquer sur « HISTORY ». Vous pouvez y apercevoir l'historique de vos transactions, et normalement il y aura la transaction que vous avez effectuée à l'étape précédente avec son statut. Vous attendez que le statut soit « Complete ».
Lorsque le statut est « Complete », cliquez sur « EXCHANGE » en haut à gauche, juste à côté de POLONIEX.

Acheter des Lumens (STR)

Vous sélectionnez l'onglet BTC à droite, juste en dessous du titre « MARKETS » et sélectionnez la ligne où il y a écrit STR.
Pour vérifier que vous avez sélectionné la bonne crypto-monnaie, vous devez apercevoir BUY STR en titre du 1er des 3 blocs en dessous du graphique comme sur l'image ci-

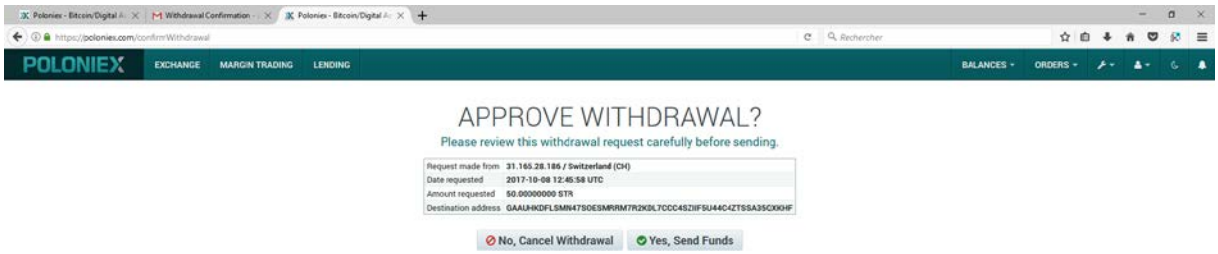
dessus. Vérifier que ce qui est écrit à coté de « You have » juste en dessous du titre BUY STR, soit bien en BTC.
Dans le champ « Amount », vous pouvez choisir combien de Lumens (STR) vous souhaitez acheter.
Dans le champ « Total », vous pouvez choisir combien de Bitcoin vous voulez utiliser pour acheter des Lumens.
Vous devez remplir qu'un de ces deux champs, l'autre champ va se remplir automatiquement. Si vous souhaitez utiliser tous vos Bitcoin pour acheter des Lumens, cliquer sur la somme à côté de « You have », juste en dessous du titre « BUY STR ». Cela va remplir tous les champs avec la somme totale de vos Bitcoin.
Une fois que vous êtes sûr de ce que vous avez rempli, cliquez sur « Buy ».
Ensuite, glisser la souris sur « BALANCES » et cliquez sur « Deposits & Withdrawals ».

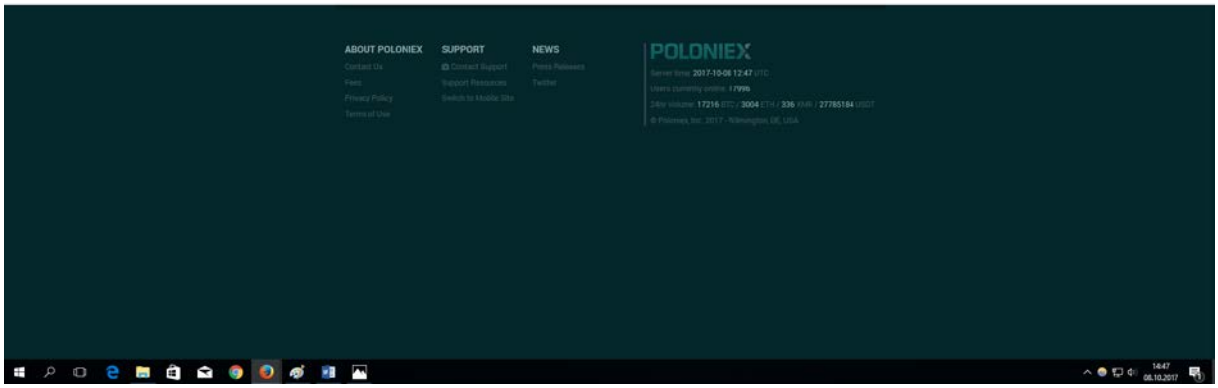
Envoyer les Lumens



Cherchez la ligne STR, pour faciliter la recherche, cliquez sur la case à cocher « Hide 0 Balances », ça va permettre de cacher toutes les crypto-monnaies que vous n'utilisez pas.
Ensuite cliquer sur « withdraw » sur la ligne STR. 3 champs vont apparaître.
Le champ « Address » doit être rempli avec votre clé publique de votre compte sur InvestProject, Attention ! Pas votre clé publique pour vous connectez à InvestProject, mais la clé publique qui vous a été fournie lorsque vous avez accepté le règlement du site.
Le champ « Amount » doit être rempli avec le nombre de Lumens que vous souhaitez envoyer au compte que vous possédez sur InvestProject.
Vous pouvez laisser le champ Memo.id vide.
Lorsque vous avez rempli les 2 champs correctement, cliquez sur « Withdraw ». Vous allez recevoir un email avec un lien où il faudra cliquer pour arriver à l'étape suivante.

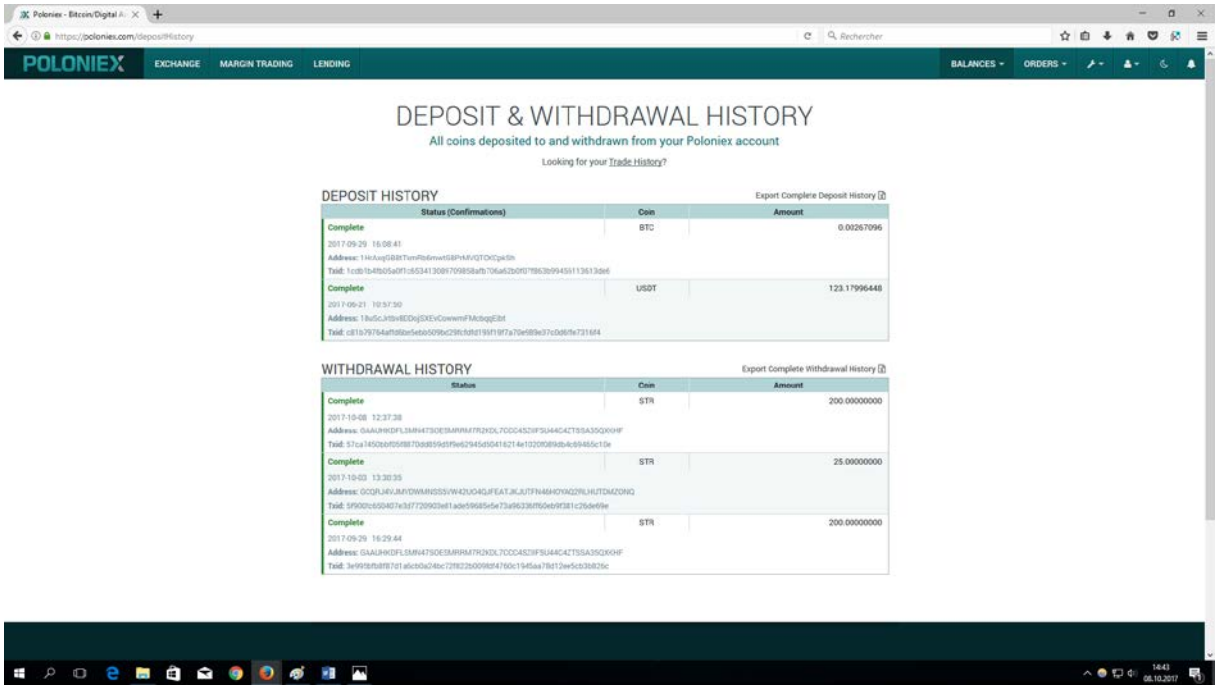
Confirmer l'envoi de Lumens





Cette page va vous montrer un résumé des données de la transaction, et va vous demander de confirmer ou non votre transaction. Si toutes les informations sont correctes, cliquer sur « Yes, Send Funds ».
Votre WITHDRAWAL va être confirmé et vous glisserez votre souris sur « BALANCES » et vous cliquerez sur « HISTORY » pour suivre votre transaction.

Voir l'historique de l'envoi de Bitcoin



Lorsque votre état de transaction sous « WITHDRAWAL HISTORY » sera « Complete », vous pourrez aller sur InvestProject et normalement, votre compte contiendra des Lumens, qui vous permettront d'investir sur les propositions de projet.

Annexe 3 : Procès-verbaux des réunions de suivi

PROCÈS VERBAL DE LA RÉUNION 01 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 01 du travail de Bachelor convoquée et tenue le vendredi 07 juillet 2017 à 14H, à l'HEG.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

Athanasios PRIFTIS – Professeur de suivi du travail de Bachelor

2. Recommandations lors de la réunion :

1. Tirer des leçons de la faille exploitée de THE DAO.
2. Mettre les descriptions dans l'observatoire sur Cowaboo.
3. Faire un fichier où je note ce que je fais à chaque fois : quels problèmes j'ai, comment je les résous, etc.)
4. Il y aura une présentation intermédiaire en Septembre, pour expliquer à Mr. Trabichet, ce qui a été fait et ce qui va être fait, et recueillir des conseils et avis de sa part.
5. Préparer la description du projet. Demander les dates du projet, l'impact du projet, qu'est-ce qu'il va faire. Pour qu'on sache quoi voter. Ça posté sur cowaboo.
6. Trouver un nom pour le projet du travail de Bachelor, un nom vendeur.
7. Priviligier le développement des transactions avec stellar, pour y avoir déjà touché et vu les problèmes et opportunités avant la présentation intermédiaire.
8. Aller sur Stellar.org.
9. Comment ils peuvent envoyer de l'argent.
10. Est-ce qu'on créer notre propre monnaie, pour les votes. Comment on va voter pour le projet ?
11. Voir pour faire un compte de gestion sur Stellar.

12. Soit les gens vont promettre de mettre de l'argent, soit avec des votes ?
13. Bien décrire toutes les règles possibles, montrer qu'on réfléchit à plusieurs scénarios.
14. Voir pour la heg propose un projet directement, en tant que super admin.
15. Une école, elle finance cette semaine 50% d'un projet proposé par un étudiant.
16. Voir que la heg peut proposer de doubler l'investissement à celui qui gagne.
17. Montrer que si la heg voit que c'est sérieux et faire un scénario où la heg prend le risque.
18. Faire une inscription pour que les gens aient un compte Stellar.
19. Voir comment les gens peuvent investir sur Stellar.
20. Aller voir le site Stellar developpers et regarder ses forums.
21. Démontrer comment j'ai recherché les solutions à mes problèmes.
22. Quand on rentre, il faut que ça créer une wallet stellar, le compte stellar créer un wallet.
23. On peut créer une monnaie qui est liée avec les lumens. Les gens voient une taxe pour Stellar et peut-être le site taxe une partie des transactions, pour payer les frais du site.
24. Implémenter la transaction automatisée. Voir la règle pour qui investit dans quoi.
25. Il faut montrer un monitoring, avec qui a voté sur le projet et avec combien d'argent. Récupérer avec le numéro de transaction, pour voir qui a voté et combien.
26. Réunion le 29 août mardi à 9 heures (réunion intermédiaire de suivi avec monsieur Trabichet) faire un PowerPoint, pour montrer la 1ère partie du projet, pour voir avec M. Trabichet avant la finalisation du code et du projet.
27. 25 Juillet 2017 à 14h Rendez-vous de suivi.
28. Définir le nom de l'application rapidement et l’envoyer par mail.

PROCÈS VERBAL DE LA RÉUNION 02 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 02 du travail de Bachelor convoquée et tenue le mardi 25 juillet 2017 à 14H30, à l'HEG.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

Athanasios Priftis – Professeur de suivi du travail de Bachelor

2. Décisions prises lors de la réunion :

1. Chaque personne qui s'inscrit sur le site, à directement un compte Cowaboo qui est lui-même un compte Stellar.
2. Regarder comment Stellar permet de payer avec plusieurs monnaies.
3. Pour le compte de gestion, voir comment récupérer la somme pour la distribuer aux comptes correspondants.
4. Bien montrer que l'argent va être récupéré soit par les investisseurs, soit par celui qui a proposé le projet.
5. Voir pour faire un règlement qui montre aux gens comment l'argent est géré par le compte du site.
6. On voit les risques dans THE DAO avec des codes trop avancées qu'on n'est peut-être pas prêt pour faire des contrats automatiques.
7. Nous on fait des contrats semi-automatiques, c'est-à-dire du code qui fait les choses automatiquement, mais qui peut être modifié par la suite.
8. Expliquer pourquoi on utilise Stellar.
9. Il faut expliquer comment les gens peuvent utiliser leur compte, tutoriel.
10. Voir les paiements avec Stellar, onglet : « send and receive money ».
11. Faire une implémentation d'une fonctionnalité qui permet de recharger son compte, mais si c'est trop compliqué, faire un document tutoriel, qui explique correctement comment recharger son compte.
12. Créer un compte sur Stellar pour faire des tests.
13. Les autres monnaies sur Stellar sont converties à chaque fois en Lumens.

14. Faire une liste des différents problèmes qui vont être codé, pour aider à développer l'application, voir ce qu'il y a à faire (un product backlog).
15. Voir l'api de Stellar.
16. Télécharger Stellar desktop client.
17. Tester le client de Stellar, pour voir ce qu'il propose.
18. Créer les comptes Stellar à l'inscription et s'en créer un soi-même pour le tester.
19. Pour les données enregistrées sur Cowaboo, faire une seul entry et y envoyer du JSON.
20. Réfléchir à un 3ème rôle en plus des investisseurs et proposant. Ce rôle sera de faire partis de la gouvernance du site. Ne pas le développer, mais en parler pour des améliorations futures.
21. Dans la gouvernance, permettre des votes qui suite à un concensus, décidera des améliorations futures du site internet.
22. Voir Stellar sur slack, pour pouvoir poser des questions à la communauté.
23. Faire la liste des stories et ensuite les mettre dans des sprints avec des tâches, pour savoir ce que je vais faire chaque semaine.
24. Réunion Skype le 16 aout 2017 à 10 heures. Envoyer un email de rappel 1 à 2 jours à l'avance.

PROCÈS VERBAL DE LA RÉUNION 03 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 03 du travail de Bachelor convoquée et tenue le mercredi 16 août 2017 à 10H, sur skype.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

Athanasios Priftis – Professeur de suivi du travail de Bachelor

2. Décisions prises lors de la réunion :

1. Mettre ce que je fais, chaque étape.
2. Mettre les choix que j'ai fait et comment je les utilises.
3. Faire des tests dans l'API de Cowaboo en postant du JSON.
4. Garder des semaines pour voir les révisions de stories.
5. Organiser les sprints sur 10 jours, les sprints sont plus intenses, c'est coder à fond pour ça.
6. Faire attention avec le texte du règlement, les personnes qui vont juger vont se mettre à la place de l'utilisateur.
7. Bien détailler chaque étape du scénario dans un Powerpoint. Il faut que Mr. Trabichet comprenne l'enjeu et ce que l'application va faire.
8. Expliquer les étapes de l'utilisateur et montrer l'application visuellement. Montrer que les projets peuvent être expliqués.
9. Créer la présentation pour que Mr. Trabichet donne son bon avis.
10. Ce n'est pas des sprints ce que je fais, c'est des périodes de travail.
11. Ne faire qu'un seul salon avec un objectif commun d'investissement.
12. Garder un historique avec les anciennes périodes d'investissement. Montrer qu'il y a un temps pour l'investissement.
13. Dans la partie paiement ajouter la partie métrique. Voir la partie monitor, combien d'argent ils ont reçu et voir quel type d'informations sont utiles pour ça dans les transactions.
14. Bien mettre le détail des investissements pour chaque personne.
15. Bien utiliser les informations que Stellar me propose.

16. Préparer des questions précises où il y a des doutes, pour les poser durant la présentation intermédiaire.
17. Voir comment l'investissement se fait.
18. La Blockchain aide à être plus transparent, car on peut voir les transactions faites.
19. Dire ce que j'ai déjà fait et expliquer ce que je veux faire pour poser mes questions.
20. Envoyer le Powerpoint avant la présentation avec Mr. Trabichet. le 28 août 2017.
21. Faire le Powerpoint d'environ 20 slide, bien d'écrire le scénario étape par étape du point de vue de l'utilisateur.

PROCÈS VERBAL DE LA RÉUNION 04 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 04 du travail de Bachelor convoquée et tenue le mardi 29 août 2017 à 9H, à l'HEG.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

Athanasios Priftis – Professeur de suivi du travail de Bachelor

Jean Philippe Trabichet – Professeur du travail de Bachelor

2. Décisions prises lors de la réunion :

1. Rendre les changements transparents pour les utilisateurs, c'est-à-dire que lorsque le site changera des choses dans sa façon de fonctionner, les utilisateurs devront être mis au courant.
2. Bien retenir ce que signifie l'acronyme DAO (Decentralised Autonom Organisation).
3. Bien expliquer au début de la présentation en quoi consiste le projet, car pour moi c'est implicite, mais les personnes qui ne connaissent pas le projet ne savent pas.
4. Être clair dans la présentation pour des personnes qui découvre le projet.
5. Si un des concepts présenté est détaillé plus tard dans la présentation, dire qu'il sera expliqué plus loin.
6. Mettre un sommaire, où le faire avec les diapos générales du Prezi.
7. Bien automatiser à la fin d'une période, le fait que l'argent soit déplacé aux bons endroits (investisseurs ou proposants).
8. Ne pas mettre sprint dans le Gantt et dans le fichier Excel, mettre période de travail à la place.
9. Prochaine réunion le vendredi 22 septembre 2017.

PROCÈS VERBAL DE LA RÉUNION 05 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 05 du travail de Bachelor convoquée et tenue le mercredi 20 Septembre 2017 à 11H, à l'HEG.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

Athanasios Priftis – Professeur de suivi du travail de Bachelor

2. Décisions prises lors de la réunion :

1. Faire le monitor.
2. Lors de la soutenance, je peux montrer l'algorithme du code qui va être exécuté à la fin d'une période.
3. Si possible, faire investir sur une des propositions, les personnes qui sont à la soutenance.
4. Expliquer qu'il manque les contrôles.
5. Prochain Rendez-vous : le mercredi 11 octobre 2017
6. Voir pour mettre le site internet en ligne.
7. Je peux montrer le Git hub pour la présentation
8. Permettre de créer des comptes avant la présentation.
9. Faire une sorte d'atelier pour que les personnes présentes ne soit pas que spectateurs, mais soit plus acteurs.
10. Dans le document, je montre des bouts de code et je mets le lien du Git hub.
11. Montrer le plus important du code fait.
12. Faire un retour sur Cowaboo. Expliquer pour la base de données.
13. Essayer d'ajouter 20 Lumens pour rendre le compte actif, à chaque création de compte.
14. Demander de venir avec des laptops à la présentation.
15. Demander de créer les comptes avant pour tester à la présentation.
16. Préparer les algorithmes importants, pour pouvoir les sortir s'il y a des questions qui les concernent.
17. Préparer l'explication du DAO.
18. Récupérer le change Lumens / CHF.
19. Faire un objectif de 3000 pour les projets.
20. Dire que le compte peut être utilisé sur d'autres applications qui utilisent Stellar.
21. Montrer l'historique des périodes.
22. Expliquer qui récupère l'argent.
23. Montrer qu'on peut changer les règles. 50% ou 100% de l'investissement.

PROCÈS VERBAL DE LA RÉUNION 06 DU TRAVAIL DE BACHELOR

Le procès-verbal de la réunion 06 du travail de Bachelor convoquée et tenue le mercredi 11 octobre 2017 à 11H, à l'HEG.

1. Membres présents à la réunion

Amir SAIDI – Etudiant effectuant le travail de Bachelor

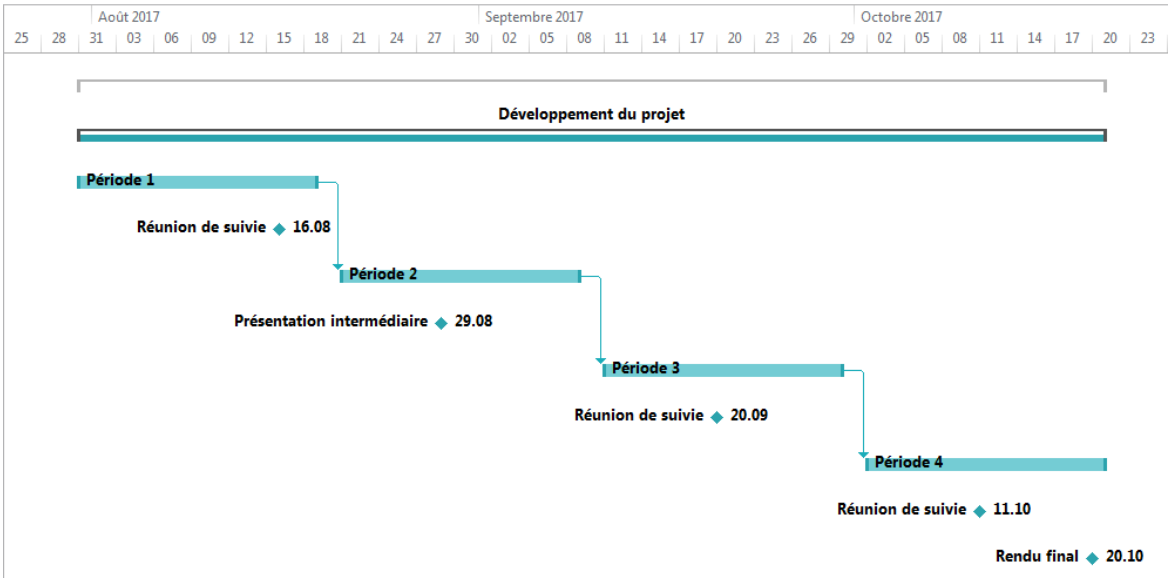
Athanasios Priftis – Professeur de suivi du travail de Bachelor

2. Décisions prises lors de la réunion :

24. Mettre les codes en liens avec les règles.
25. Il n'y a pas besoins des 2 comptes sur le site, le compte Cowaboo est un compte Stellar et donc on peut faire les transactions avec. Donc, modifier le code pour utiliser uniquement le compte Cowaboo que les personnes ont à l'inscription.
26. Voir Apps.cowaboo.net pour tester les transactions avec le compte Cowaboo.
27. Mettre les procès-verbaux et la planification en annexe du document de travail de Bachelor.
28. Enlever les deuxièmes comptes, juste avec le compte Cowaboo ça fonctionne.
29. Voir pour l'hébergement du site avec Nodejs après la reddition, si ça marche faire 3-4 comptes avec quelques lumens pour faire tester aux personnes durant la présentation, sinon faire la démonstration moi-même en local.
30. Faire une introduction avec un Powerpoint, en expliquant ce qui m'a inspiré, l'idée et ensuite montrer l'exemple sur le site. Ensuite, expliquer que le 1er du mois suivant à 00h l'algorithme va décider de si l'argent va retourner aux investisseurs ou au proposant. Montrer l'algorithme et l'expliquer.
31. Préparer les bouts de codes en rapport avec les règles pour montrer quel bout de code correspond à quelle règle.
32. Dans le règlement affiché sur le site, faire un tableau avec la règle écrite et dans la colonne à côté, la règle définie en code.
33. Rendre un seul document. Expliquer le code important (algorithme automatique, investissement, ajout des propositions sur Cowaboo, inscription avec Cowaboo.
34. Faire un petit manuel d'utilisateur pour le site, en le mettant en annexe.

Annexe 4 : Plannification

Diagramme de Gant



Liste des stories

Id	En tant que...	Je souhaite...	Priorité	Pts de difficultés	Statut
S01	Développeur	Faire la description du projet et du site internet	Haute	5	✓
S02	Développeur	M'informer sur THE DAO et analyser son fonctionnement, ainsi que la faille qui a été exploitée	Haute	10	✓
S03	Développeur	Lister les différents scénarios possibles, afin de choisir lequel va être développé	Haute	8	✓
S04	Investisseur	Investir de l'argent sur des propositions de projet	Haute	30	✓
S05	Développeur	Créer et associer un compte aux utilisateurs, lors de leur inscription	Haute	50	✓
S06	Investisseur	Alimenter mon compte avec de l'argent	Haute	50	✓
S07	Utilisateur	Avoir un aperçu de la balance de mon compte	Haute	25	✓
S08	Investisseur	Lire la description de la proposition	Moyenne	15	✓
S09	Proposant	Créer une proposition avec sa description	Moyenne	20	✓
S10	Utilisateur	Lire le règlement pour m'assurer de la bonne foie du site internet	Moyenne	15	✓
S11	Utilisateur	Lire la présentation du site internet et de son but	Moyenne	10	✓
S12	Développeur	Faire la mise en page (HTML/CSS) du site internet	Moyenne	25	✓
S13	Développeur	Faire la documentation du rendu	Moyenne	15	✓
S15	Utilisateur	Avoir une vision des transactions faites pour une proposition	Faible	20	✓
S16	Utilisateur	Proposer des changements pour le site internet	Faible	30	A faire
S17	Utilisateur	Voter pour une proposition de changement du site internet	Faible	20	A faire

Periode 1

ID	Stories	Tâches	Statut	Statut stories
S04	Investir de l'argent sur des propositions de projet	Créer deux comptes sur stellar	Fait	✓
		Alimenter les comptes pour faire des tests	Reporté	
		Se renseigner sur le fonctionnement des transactions et sur les forums	Fait	
		Développer et tester la fonctionnalité	Fait	
S05	Créer et associer un compte aux utilisateurs, lors de leur inscription	Se renseigner sur la création de comptes par code sur les sites et forums	Fait	→
		Développer et tester la fonctionnalité	Reporté	
		Prévoir un plan B en cas de difficultés acrues	Non fait	

Remarques :
S04 : , les tests ont pu être fait avec la monnaie native, chaque compte en a 10'000 à la création. Donc, pour l'alimentation du compte, cela concerne plus le prochain sprint, où il y a la story concernant l'alimentation de son compte.
Donc, vu que les tests ont pu être fait, la story est terminée.
S05 : Il y a plusieurs difficultés pour les 2 stories, car lors du renseignement sur le code pour faire les transactions et les comptes, il a été vu que les codes devait être utilisés avec node.js et donc il y a eu un apprentisage par des tutoriels de node.js. De plus, il y a eu également un peu de perte de temps, sur l'installation et l'apprentissage de l'utilisation de GITHUB et GIT, pour assurer une sécurité du code en cas de problème.
A cause de ses problèmes, il y eu du retard, et donc cette story a été reporté sur le sprint suivant pour la terminer.

Periode 2

ID	Stories	Tâches	Statut	Statut stories
S05	Créer et associer un compte aux utilisateurs, lors de leur inscription	Faire le JSON des comptes des utilisateurs.	Fait	✓
		Tester la création du compte et l'ajout dans le fichier JSON	Fait	
S06	Alimenter mon compte avec de l'argent	Se renseigner sur les sites internets et les forums	Fait	→
		Développer et tester la fonctionnalité	Echec	
		Prévoir un plan B en cas de difficultés acrues	Reporté	
S07	Avoir un aperçu de la balance de mon compte	Se renseigner sur les sites internets et les forums	Fait	✓
		Développer et tester la fonctionnalité	Fait	
		Prévoir un plan B en cas de difficultés acrues	Non fait	

Remarques :
La story S06 a été reporté, car je pensais que le plan B, qui est faire un tutoriel pour alimenter son compte avec du bitcoin depuis un autre site, serait rapide à faire. Donc, j'ai essyé longtemps de trouver une solution pour intégrer cette fonctionnalitée dans le site, car je trouve cela plus intéressant. Lorsque je suis passé au plan B, j'ai recherché pendant plusieurs jours comment mettre des bitcoins sur un compte stellar et j'ai eu des réponses de personnes sur le slack stellar public, qui ne correspondait pas à ma demande, mais je m'en rendait compte après avoir essayé plusieurs choses. Ensuite, comme je perdais beaucoups de temps à essayer de trouver comment faire, j'ai décider d'arrêter, car la prochaine période avait déjà commencée et du coup, j'ai commencé à faire la partie proposition qui est également une partie importante, donc j'ai reporté la partie alimenter mon compte avec de l'argent dans la période suivante et si je n'y arrive toujours pas, ou je n'ai plus le temps, je partirai sur le plan B qui est : faire marcher le site en utilisant la monnaie native qui est fournie lors de la création d'un compte sur le horizon-testnet.

Periode 3

ID	Stories	Tâches	Statut	Statut stories
S06	Alimenter mon compte avec de l'argent	Se renseigner sur les sites internets et les forums	Fait	✓
		Alimenter un compte essai pour vérifier que ça fonctionne	Fait	
		Faire le tutoriel pour les utilisateurs	Fait	
		Mettre le pdf à disposition sur le site internet	Fait	
S08	Lire la description de la proposition	Faire une entry avec un JSON contenant des exemples tests	Fait	✓
		Récupérer le JSON et sélectionner une proposition particulière	Fait	
		Afficher la proposition	Fait	
S09	Créer une proposition avec sa description	Créer le formulaire	Fait	✓
		Intégrer un éditeur de texte Pour la description du projet	Fait	
		Faire l'ajout dans le JSON des propositions	Fait	
		Tester l'ajout et l'affichage de la proposition	Fait	
		Faire la mise en page	Fait	
S10	Lire le règlement pour m'assurer de la bonne foie du site internet	Ecrire le réglement du site internet	Fait	✓
		Faire les screen du code qui met en place la règle et le lier à la règle.	Fait	
		Vérifier l'orthographe	Fait	
S11	Lire la présentation du site internet et de son but	Faire une explication du site internet et de son but	Fait	✓
		Vérifier l'orthographe	Fait	

Remarques :
Tutoriel pour alimenter son compte en Lumens réussi, donc les investissement ne se feront pas en bitcoin, mais ne se feront pas également avec la monnaie de test de horizon testnet. Donc ce sera du vrai argent qui sera utilisé.

Periode 4

ID	Stories	Tâches	Statut	Statut stories
S15	Avoir une vision des transactions faite pour une proposition	Se renseigner sur les sites internets et les forums	Fait	✓
		Développer et tester la fonctionnalité	Fait	
S12	Faire la mise en page (HTML/CSS) du site internet	Mettre en place un thème pour toutes les pages	Fait	✓
		Faire le menu de navigation	Fait	
		Parcourir tout le site internet pour voir que tout a été modifié	Fait	
S13	Faire la documentation du rendu	Faire le document à rendre du travail de bachelor	Fait	✓
		Ajouter les autres documents en annexe	Fait	
		Vérifier l'orthographe de tous les documents	Fait	
		Faire la mise en page et générer le pdf	Fait	